A. HARRY

Report Name: KEYDIS

Date: 01-Apr-1982

Time: 14:09:20

Job Number: 00990

Location Name: #DEFAULT PRN1

```
**********************************************************
*                                                        *
*       Report Name:  KEYDIS                             *
*                                                        *
*       Date:  01-Apr-1982                               *
*                                                        *
*       Time:  14:09:20                                  *
*                                                        *
*       Job Number:  00990                               *
*                                                        *
*       Location Name:  #DEFAULT.PRN1                    *
*                                                        *
*                                                        *
*                                                        *
**********************************************************

ERR LINE  ADDR  01 02 03 04        SWEET16 EQUATES                    PAGE    1
```

```
EAR LINE  ADDR  01 02 03 04        SWEET16 EQUATES                                  PAGE    1

      421                               LIST    B,W
      422                               TITLE   'DISPLAY HANDLER  -- 10-00-78 --  DISPLO'
      423
      424                       ; SOURCE FILE = KEYDIS
      425                       ;
      426                               NOLIST  A
      427                               NAME    KEYDIS
      428                       ;
      429                               PSCT
      430                       ;
      431                               XDEF    PWRONA,RETURJ,NOFUNC
      432                               XDEF    DOPEN,GETCH,OUTCH,DRAW
      433                               XDEF    EOPEN,EGETCH,EOUTCH
      434                               XDEF    KGETCH
      435                               XDEF    PIRQ5
      436                       ;
      437                               XREF    CH,MEMTOP,SHFLOK,CHBAS
      438                               XREF    CHACT,SDMCTL,TINDEX,CRSINH
      439                               XREF    TABMAP,COLOR0,TXTMSC,GPRIOR
      440                               XREF    COLOR4,BOTSCR,SDLSTL
      441                               XREF    TXTROW,TXTCOL,DMASK,CHAR
      442                               XREF    ATACHR,SSFLAG,ESCFLG,DSPFLG
      443                               XREF    INVFLG,LOGMAP,SCRFLG,SUBTMP
      444                               XREF    HOLD2,TMPLBT,HOLD3
      445                               XREF    TMPROW,FILFLG,HOLD4,CH1
      446                               XREF    KEYDEL,SRTIMR,FILDAT,NOCLIK
      447                               XREF    HELPFG,DMASAV,SUPERF,FINE
      448                               XREF    NEWROW,NEWCOL,ROWINC,COLINC
      449                               XREF    KRPDEL,VSFLAG,KEYDIS,RAMSIZ
      450                       ;
      451                               XREFB   RAMTOP,ICAX2Z,ICAX1Z,DINDEX
      452                               XREFB   DSTAT,ROWMSK,ADRESS,SWPFLG
      453                               XREFB   HOLD1,DFNTMP,SAVMSC,SAVADR
      454                               XREFB   LMARGN,RMARGN,ROWCRS,SHFAMT
      455                               XREFB   OLDROW,TMPCHR,OLDCHR,BUFCNT
      456                               XREFB   BUFSTR,COLCRS,LOGCOL,RRKEY
      457                               XREFB   HOLDCH,INSDAT,FRMADR,TOADR
      458                               XREFB   APPMHI,ALTTMP,OLDADR,ROWAC
      459                               XREFB   ENDPT,BITMSK,ICCOMZ
      460                               XREFB   DELTAR
      461                               XREFB   OLDCOL,DELTAC,COLAC
      462                               XREFB   COUNTR,ATRACT,KEYDEF,FKDEF
      463                       ;
      464                       ;
      465  0000  A9 FF          PWRONA  LDA     #$FF
      466  0002  8D 00 00     E          STA     CH
      467  0005  AD 00 00     E          LDA     RAMSIZ
      468  0008  85 00        E          STA     RAMTOP
      469  000A  A9 40                   LDA     #$40            ;DEFAULT TO UPPER CASE ALPHA AT PWRON
      470  000C  8D 00 00     E          STA     SHFLOK
      471  000F  A9 97        F          LDA     #<LOW.ATASCII   ;SETUP KEYBOARD DEFINITION TABLE POINTER
      472  0011  85 00        E          STA     KEYDEF
      473  0013                          LDA#H   ATASCII
      474  0015  85 01        E          STA     KEYDEF+1
```

```
     479  0017  A9 57        P          LDA     # LOW.FUNCKY    ;SETUP FUNCTION KEY DEF TABLE POINTER
     480  0019  85 00        E          STA     FKDEF
     481  001B                         LDAIH   FUNCKY
     486  001D  85 01        E          STA     FKDEF+1
     487  001F  60                     RTS                     ;POWER ON COMPLETED
```

```
                        PAGE
                ;
                ; BEGIN DISPLAY HANDLER OPEN PROCESSING
                ;
0020 A5 00      E DOPEN   LDA     ICAX2Z      ;GET AUX 2 BYTE
0022 29 1F              AND     #$1F
                         IF      FALSE       ;ALLOW MODE 0 SPLIT SCREEN.
                        BNE     OPNCOM      ;IF MODE ZERO, CLEAR ICAX1Z
                         ELSE
0024 4C 2F 00   P       JMP     OPNCOM
                         ENDIF
0027 A5 00      E EOPEN1  LDA     ICAX1Z      ;CLEAR "CLR INHIBIT" AND "MXD MODE" BITS
0029 29 0F              AND     #$F
002B 85 00      E       STA     ICAX1Z
002D A9 00              LDA     #0
002F 85 00      E OPNCOM  STA     DINDEX
0031 C9 11              CMP     #17
0033 90 05              BCC     DOPEN6      ;MODE < 17, IO = 161.
0035 A9 91              LDA     #BADMOD
0037 4C E1 01   P       JMP     DERR
003A A9 E0      DOPEN6  LDA     #DOMCHR     ; INITIALIZE GLOBAL VBLANK RAM
003C 8D 00 00   E       STA     CHBAS
                ;
003F A9 02              LDA     #2
0041 8D 00 00   E       STA     CHACT
0044 8D 00 00   E       STA     SDMCTL      ;TURN OFF DMA NEXT VBLANK
                ;
0047 A9 01              LDA     #SUCCES
0049 85 00      E       STA     DSTAT       ;CLEAR STATUS
                ;
004B A9 C0              LDA     #$C0        ;DO IRGEN
004D 05 00      E       ORA     POKMSK
004F 85 00      E       STA     POKMSK
0051 8D 0E D2           STA     IRGEN
                ;
0054 A9 40              LDA     #$40        ;IF FINE SCROLLING ENABLE DLI
0056 2C 00 00   E       BIT     FINE
0059 10 02              BPL     NOFINE      ;NOT FINE SCROLLING (VBLANK ONLY).
005B A9 C0              LDA     #$C0
005D 8D 0E D4   NOFINE  STA     NMIEN
                ;
0060 A9 00              LDA     #0
0062 8D 00 00   E       STA     TINDEX      ;TEXT INDEX MUST ALWAYS BE 0
0065 85 00      E       STA     ADRESS
0067 85 00      E       STA     SWPFLG
0069 8D 00 00   E       STA     CRSINH      ;TURN CURSOR ON AT OPEN
                ;
006C A0 0E              LDY     #14         ;CLEAR TAB STOPS
006E A9 01              LDA     #1          ;INIT TAB STOPS TO EVERY 8 CHARACTERS
0070 99 00 00   E CLRTBS  STA     TABMAP,Y
0073 88                 DEY
0074 10 FA              BPL     CLRTBS
                ;
0076 A2 04              LDX     #4          ;LOAD COLOR REGISTERS
```

```
0079  BD 4E 0C     P  DOPENB  LDA   COLRTB,X
007B  9D 00 00     E          STA   COLOR0,X
007E  CA                      DEX
007F  10 F7                   BPL   DOPENB
                      ;
0081  A4 00        E          LDY   RAMTOP     ;DO TXTMSC=$3F60 (IF MEMTOP=4000)
0083  88                      DEY
0084  8C 01 00     E          STY   TXTMSC+1
0087  A9 60                   LDA   #$60
0089  8D 00 00     E          STA   TXTMSC
                      ;
008C  A6 00        E          LDX   DINDEX
008E  BD 05 0B     P          LDA   ANCONV,X   ;CONVERT IT TO ANTIC CODE
0091  85 00        E          STA   HOLD1
0093  A5 00        E          LDA   RAMTOP     ;SET UP AN INDIRECT POINTER
0095  85 01        E          STA   ADRESS+1
0097  BC A2 0B     P          LDY   ALOCAT,X   ;ALLOCATE N BLOCKS OF 40 BYTES
009A  A9 28           DOPEN1  LDA   #40        ;ADRESS = ADRESS-(40*N)
009C  20 30 06     P          JSR   DBSUB
009F  88                      DEY
00A0  D0 F8                   BNE   DOPEN1
                      ;
00A2  AD 00 00     E          LDA   GPRIOR     ;CLEAR GTIA MODES
00A5  29 3F                   AND   #$3F
00A7  85 01        E          STA   OPNTMP+1
00A9  A8                      TAY
00AA  E0 08                   CPX   #8         ;TEST IF 320X1
00AC  90 1F                   BCC   NOT8       ;MODE < 8
00AE  E0 10                   CPX   #16
00B0  F0 0D                   BEQ   DOP030     ;MODE = 16
00B2  E0 0C                   CPX   #12
00B4  B0 17                   BCS   NOT8       ;MODE >= 12
00B6  8A                      TXA              ;GET 2 LOW BITS
00B7  6A                      ROR   A
00B8  6A                      ROR   A
00B9  6A                      ROR   A
00BA  29 C0                   AND   #$C0       ;NOW 2 TOP BITS
00BC  05 01        E          ORA   OPNTMP+1
00BE  A8                      TAY
00BF  A9 10           DOP030: LDA   #16        ;SUBTRACT 16 MORE FOR PAGE BOUNDARY
00C1  20 30 06     P          JSR   DBSUB      ;WE WANT A LINE BOUNDARY AT X000
00C4  E0 0B                   CPX   #11        ;TEST MODE 11
00C6  D0 05                   BNE   NOT8       ;IF MODE = 11
00C8  A9 06                   LDA   #6         ;PUT GTIA LUM VALUE INTO BACKGROUND REGISTER
00CA  8D 00 00     E          STA   COLOR4
00CD  8C 00 00     E  NOT8:   STY   GPRIOR     ;STORE NEW PRIORITY
                      ;
00D0  A5 00        E          LDA   ADRESS     ;SAVE MEMORY SCAN COUNTER ADDRESS
00D2  85 00        E          STA   SAVMSC
00D4  A5 01        E          LDA   ADRESS+1
00D6  85 01        E          STA   SAVMSC+1
                      ;
00D8  AD 0B D4        VBWAIT: LDA   VCOUNT     ;WAIT FOR NEXT VBLANK BEFORE MESSING
00DB  C9 7A                   CMP   #$7A       ;WITH THE DISPLAY LIST
```

```
 596  0DDD  D0 F9                    BNE     VBWAIT
 597
 598  0DDF  20 2E 06        P        JSR     DBDEC           ;START PUTTING DISPLAY LIST RIGHT UNDER RAM
 599
 600  0DE2  BD E6 08        P        LDA     PAGETB,X        ;TEST IF DISPLAY LIST WILL BE IN TROUBLE
 601  0DE5  F0 06                    BEQ     NOMOD           ;OF CROSSING A 256 BYTE PAGE BOUNDARY
 602  0DE7  A9 FF                    LDA     #$FF            ;IF SO, DROP DOWN A PAGE
 603  0DE9  85 00           E        STA     ADRESS
 604  0DEB  C6 01           E        DEC     ADRESS+1
 605
 606  0DED  20 22 06        P NOMOD  JSR     DBUDEC          ;(DOUBLE BYTE DOUBLE DECREMENT)
 607  0DF0  A5 00           E        LDA     ADRESS          ;SAVE END OF DISPLAY LIST FOR LATER
 608  0DF2  85 00           E        STA     SAVADR
 609  0DF4  A5 01           E        LDA     ADRESS+1
 610  0DF6  85 01           E        STA     SAVADR+1
 611
 612  0DF8  A9 41                    LDA     #$41            ;(ANTIC) WAIT FOR VBLANK AND JMP TO TOP
 613  0DFA  20 26 06        P        JSR     STORE
 614  0DFD  86 00           E        STX     OPNTMP
 615  0DFF  A9 18                    LDA     #24             ;INITIALIZE BOTSCR
 616  0101  8D 00 00        E        STA     BOTSCR
 617  0104  A5 00           E        LDA     DINDEX          ;DISALLOW MIXED MODE IF MODE 9-11,
 618  0106  C9 0C                    CMP     #12             ;... TEXT IS UNREADABLE
 619  0108  B0 04                    BCS     MIXOK           ;MODE >= 12
 620  010A  C9 09                    CMP     #9
 621  010C  B0 41                    BCS     NOTMXD          ;MODE >= 9
 622  010E  A5 00           E MIXOK  LDA     ICAX1Z          ;TEST MIXED MODE
 623  0110  29 10                    AND     #$10
 624  0112  F0 3B                    BEQ     NOTMXD
 625  0114  A9 04                    LDA     #4
 626  0116  8D 00 00        E        STA     BOTSCR
 627  0119  A2 02                    LDX     #2              ;ADD 4 LINES OF TEXT AT BOTTOM OF SCREEN
 628  011B  AD 00 00        E        LDA     FINE            ;FINE SCROLLING?
 629  011E  F0 08                    BEQ     DOPEN2          ;NO
 630
 631  0120  A9 02                    LDA     #$02            ;YES -- ADD EXTRA LINE
 632  0122  20 26 06        P        JSR     STORE
 633  0125  A9 A2                    LDA     #$A2            ;DLI ON LAST VISIBLE LINE
 634  0127  20 26 06        P        JSR     STORE
 635  012A  CA                       DEX
 636
 637  012B  A9 22           DOPEN2   LDA     #$22
 638  012D  20 26 06        P        JSR     STORE
 639  0130  CA                       DEX
 640  0131  10 F8                    BPL     DOPEN2
 641
 642  0133  A4 00           E        LDY     RAMTOP          ;RELOAD MSC FOR TEXT
 643  0135  88                       DEY
 644  0136  98                       TYA
 645  0137  20 26 06        P        JSR     STORE
 646  013A  A9 60                    LDA     #$60
 647  013C  20 26 06        P        JSR     STORE
 648  013F  A9 42                    LDA     #$42
 649  0141  20 26 06        P        JSR     STORE
```

```
0144  18               CLC
0145  A9 11            LDA   #MXDMOD-NUMDLE ;POINT X AT MIXED MODE TABLE
0147  65 00      E     ADC   OPNTMP
0149  A8               TAY
014A  BE 83 06   P     LDX   NUMDLE,Y
014D  D0 19            BNE   DOPEN3          ;(JMP)
                ;
014F  A4 00      E  NOTMXD  LDY   OPNTMP
0151  BE 93 06   P     LDX   NUMDLE,Y    ;GET NUMBER OF DISPLAY LIST ENTRIES
0154  A5 00      E     LDA   DINDEX      ;MODE 0?
0156  D0 10            BNE   DOPEN3      ;NO
0158  AD 00 00   E     LDA   FINE        ;FINE SCROLLING?
015B  F0 08            BEQ   DOPEN3      ;NO
                ;
015D  A9 02            LDA   #$02        ;YES -- ADD EXTRA LINE
015F  20 26 06   P     JSR   STORE
0162  A9 A2            LDA   #$A2        ;DLI ON LAST VISIBLE LINE.
0164  20 26 06   P     JSR   STORE
0167  CA               DEX
                ;
0168  A5 00      E  DOPEN3: LDA   HOLD1       ;STORE N DLE'S
016A  20 26 06   P     JSR   STORE
016D  CA               DEX
016E  D0 F8            BNE   DOPEN3
                ;
0170  A5 00      E     LDA   DINDEX      ;DO THE MESSY 320X1 PROBLEM
0172  C9 08            CMP   #8          ;MODE < 8
0174  90 26            BCC   DOPEN5
0176  C9 10            CMP   #16
0178  F0 04            BEQ   BIGGUN      ;MODE = 16
017A  C9 0C            CMP   #12
017C  B0 1E            BCS   DOPEN5      ;MODE >= 12
017E  A2 5D         BIGGUN: LDX   #93         ;GET REMAINING NUMBER OF DLE'S
0180  A5 00      E     LDA   RAMTOP      ;RELOAD MEMORY SCAN COUNTER
0182  38               SEC
0183  E9 10            SBC   #$10        ;START 4K BELOW RAMTOP
0185  20 26 06   P     JSR   STORE
0188  A9 00            LDA   #0
018A  20 26 06   P     JSR   STORE
018D  A5 00      E     LDA   HOLD1       ;(ANTIC) RELOAD MSC CODE
018F  09 40            ORA   #$40
0191  20 26 06   P     JSR   STORE
                ;
0194  A5 00      E  DOPEN4: LDA   HOLD1       ;DO REMAINING DLE'S
0196  20 26 06   P     JSR   STORE
0199  CA               DEX
019A  D0 F8            BNE   DOPEN4
                ;
019C  A5 01      E  DOPEN5: LDA   SAVMSC+1    ;POLISH OFF DISPLAY LIST ...
019E  20 26 06   P     JSR   STORE       ; ... WITH LMS.
01A1  A5 00      E     LDA   SAVMSC
01A3  20 26 06   P     JSR   STORE
01A6  A5 00      E     LDA   HOLD1
01A8  09 40            ORA   #$40
```

```
 704  01AA  20 26 06    P          JSR   STORE
 705                    ;
 706  01AD  A9 70       E          LDA   #$70        ;24 BLANK LINES
 707  01AF  20 26 06    P          JSR   STORE
 708  01B2  A9 70       E          LDA   #$70
 709  01B4  20 26 06    P          JSR   STORE
 710  01B7  A5 00       E          LDA   ADRESS      ;SAVE DISPLAY LIST ADDRESS
 711  01B9  8D 00 00    E          STA   SDLSTL
 712  01BC  A5 01       E          LDA   ADRESS+1
 713  01BE  8D 01 00    E          STA   SDLSTL+1
 714  01C1  A9 70       E          LDA   #$70        ;ADD LAST BLANK LINE ENTRY
 715  01C3  20 26 06    P          JSR   STORE       ;POSITION ADRESS=SDLSTL-1
 716                    ;
 717  01C6  A5 00       E          LDA   ADRESS      ;STORE NEW MEMTOP
 718  01C8  8D 00 00    E          STA   MEMTOP
 719  01CB  A5 01       E          LDA   ADRESS+1
 720  01CD  8D 01 00    E          STA   MEMTOP+1
 721                    ;
 722  01D0  A0 01       E          LDY   #1          ;STORE DL JVB ADDRESS.
 723  01D2  AD 00 00    E          LDA   SDLSTL
 724  01D5  91 00       E          STA   (SAVADR),Y
 725  01D7  C8          E          INY
 726  01D8  AD 01 00    E          LDA   SDLSTL+1
 727  01DB  91 00       E          STA   (SAVADR),Y
 728                    ;
 729  01DD  A5 00       E          LDA   DSTAT       ;IF ERROR OCURRED ON ALLOCATION, OPEN THE ED
 730  01DF  10 07       E          BPL   DOPEN9
 731  01E1  48          E  DERR.   PHA               ;SAVE STATUS
 732  01E2  20 27 00    P          JSR   EOPEN       ;OPEN THE EDITOR (RECURSIVE CALL')
 733  01E5  68          E          PLA               ;RESTORE STATUS
 734  01E6  A8          E          TAY               ;AND RETURN IT TO CIO
 735  01E7  60          E          RTS
 736                    ;
 737  01E8  A5 00       E  DOPEN9  LDA   ICAX1Z      ;TEST CLEAR INHIBIT BIT
 738  01EA  29 20       E          AND   #$20
 739  01EC  D0 0B       E          BNE   DOPEN7
 740  01EE  20 D4 04    P          JSR   CLRSCR      ;CLEAR SCREEN
 741  01F1  8D 00 00    E          STA   TXTROW      ;AND HOME TEXT CURSOR (AC IS ZERO)
 742  01F4  A5 00       E          LDA   LMARGN
 743  01F6  8D 00 00    E          STA   TXTCOL
 744  01F9  A9 22       E  DOPEN7  LDA   #$22        ;EVERYTHING ELSE IS SET UP'
 745  01FB  0D 00 00    E          ORA   SDMCTL      ;SO TURN ON DMACTL
 746  01FE  8D 00 00    E          STA   SDMCTL
 747  0201  4C BF 02    P          JMP   RETUR2
 748                    ;
 749
 750  0204  20 74 07    P  GETCH:  JSR   RANGE       ;GETCH DOES INCRSR, GETPLT DOESN'T
 751  0207  20 13 02    P          JSR   GETPLT
 752  020A  20 14 08    P          JSR   INATAC      ;CONVERT INTERNAL CODE TO ATASCII
 753  020D  20 B4 06    P          JSR   INCRSR
 754  0210  4C A2 02    P          JMP   RETUR1
 755                    ;
 756  0213  20 56 06    P  GETPLT: JSR   CONVRT      ;CONVERT ROW/COLUMN TO ADRESS
 757  0216  B1 00       E          LDA   (ADRESS),Y
```

```
                        ;
                        ;
                        ;
0238  00 00 00      E   OUTCH   STA   ATACHR
                        ;       JSR   OFFCRS
0239  C9 7D               ;     CMP   #CLS        ;CLEAR SCREEN?
023B  D0 03               ;     BNE   OUTCH8      ;NO
023D  20 B4 04      P     ;     JSR   CLRSCR      ;YES
0240  4C 8F 02      P    ;     JMP   RETUR2
0243  20 74 07      P    OUTCH8  JSR   RANGE
0246  AD 00 00      E    ;     LDA   ATACHR      ;TEST FOR CARRIAGE RETURN
0249  C9 9B               ;     CMP   #CR
024B  D0 06               ;     BNE   OUTCH8
024D  20 0B 07      P     ;     JSR   DOCRWS      ;DO CR
0250  4C 8F 02      P     ;     JMP   RETUR2
0253  20 4E 02      P    OUTCH8  JSR   OUTPLT
0256  20 88 06      P     ;     JSR   INCRSR
0259  4C 8F 02      P     ;     JMP   RETUR2
                        ;
                        ;
025C  AD 00 00      E    OUTPLT  LDA   SSFLAG      ;*****LOOP HERE IF START/STOP FLAG IS NON-0
025F  D0 FB               ;     BNE   OUTPLT
0261  A2 02               ;     LDX   #2
0263  B5 00         E    CRLOOP  LDA   ROWCRS,X    ;SAVE CURSOR LOCATION FOR DRAW LINE TO DRAW
0265  95 00         E    ;     STA   OLDROW,X
0267  CA                  ;     DEX
0268  10 F9               ;     BPL   CRLOOP
026A  AD 00 00      E     ;     LDA   ATACHR      ;CONVERT ATASCII(ATACHR) TO INTERNAL(CHAR)
026D  AB                  ;     TAY               ;SAVE ATACHR
026E  2A                  ;     ROL   A
026F  2A                  ;     ROL   A
0270  2A                  ;     ROL   A
0271  2A                  ;     ROL   A
0272  29 03               ;     AND   #3
0274  AA                  ;     TAX               ;X HAS INDEX INTO ATAINT
0275  98                  ;     TYA               ;RESTORE ATACHR
0276  29 9F               ;     AND   #$9F        ;STRIP OFF COLUMN ADDRESS
0278  1D 9F 0C      P     ;     ORA   ATAINT,X    ;OR IN NEW COLUMN ADDRESS
027B  8D 00 00      E    OUTCH2  STA   CHAR
027E  20 56 06      P     ;     JSR   CONVRT
0281  AD 00 00      E     ;     LDA   CHAR
0284  46 00         E    SHIFTU  LSR   SHFAMT      ;SHIFT UP TO PROPER POSITION
0286  B0 04               ;     BCS   SHIFT2
0288  0A                  ;     ASL   A
0289  4C 84 02      P     ;     JMP   SHIFTU
028C  2D 00 00      E    SHIFT2  AND   DMASK
```

```
 812  0281  85 00        E          STA     TMPCHR      ;SAVE SHIFTED DATA
 813  0283  AD 00 00     E          LDA     DMASK       ;INVERT MASK
 814  0286  49 FF                   EOR     #$FF
 815  0288  31 00        E          AND     (ADRESS),Y  ;MASK OFF OLD DATA
 816  028A  05 00        E          ORA     TMPCHR      ;OR IN NEW DATA
 817  028C  91 00        E          STA     (ADRESS),Y
 818  028E  60                      RTS
 819                   ;
 820                   ;
 821  028F  20 13 D2     P  RETUR2  JSR     GETPLT      ;DO CURSOR ON THE WAY OUT
 822  0292  85 00        E          STA     OLDCHR
 823  0294  A6 00        E          LDX     DINDEX      ;GRAPHICS HAVE INVISIBLE CURSOR
 824  0296  00 0A                   BNE     RETUR1
 825  0298  AE 00 00     E          LDX     CRSINH      ;TEST CURSOR INHIBIT
 826  029B  00 05                   BNE     RETUR1
 827  029D  49 80                   EOR     #$80        ;TOGGLE MSB
 828  029F  20 6D D2     P          JSR     OUTCH2      ;DISPLAY IT
 829  02A2  A4 00        E  RETUR1  LDY     DSTAT       ;RETURN TO CIO WITH STATUS IN Y
 830  02A4  A9 01                   LDA     #SUCCES
 831  02A6  85 00        E          STA     DSTAT       ;SET STATUS= SUCCESSFUL COMPLETION
 832  02A8  AD 00 00     E          LDA     ATACHR      ;PUT ATACHR IN AC FOR RETURN TO CIO
 833  02AB  60           NOFUNC     RTS                 ;(NON-EXISTENT FUNCTION RETURN POINT)
 834
 835                   ; END OF DISPLAY HANDLER
 836                   ;
```

```
                        PAGE
                ;
                ; BEGIN OF SCREEN EDITOR
                ;
02A2 20 09 0A   P  EGETCH  JSR     SWAP
02A5 20 B5 07   P          JSR     ERANGE
02A8 A5 00      E          LDA     BUFCNT      ;ANYTHING IN THE BUFFER?
02AA D0 3A      E          BNE     EGETC3      ;YES
02AC A5 00      E          LDA     ROWCRS      ;NO. SO SAVE BUFFER START ADDRESS
02AE 85 00      E          STA     BUFSTR
02B0 A5 00      E          LDA     COLCRS
02B2 85 01      E          STA     BUFSTR+1
02B4 20 BF 02   P  EGETC1  JSR     #GETCH ;LET'S FILL OUR BUFFER
02B7 84 00      E          STY     DSTAT       ;SAVE KEYBOARD STATUS
02B9 AD 00 00   E          LDA     ATACHR      ;TEST FOR CR
02BC C9 9B                 CMP     #CR
02BE F0 12                 BEQ     EGETC2
02C0 20 20 03   P          JSR     DOSS        ;NO, SO PRINT IT
02C3 20 00 0A   P          JSR     SWAP        ;JSR DOSS DID SWAP SO SWAP BACK
02C6 A5 00      E          LDA     LOGCOL      ;BEEP IF NEARING LOGICAL COL 120
02C8 C9 71                 CMP     #113
02CA D0 03                 BNE     EGETC6
02CC 20 06 0A   P          JSR     BELL
02CF 4C BF 02   P  EGETC6  JMP     EGETC1
02D2 20 C2 07   P  EGETC2  JSR     OFFCRS      ;GET BUFFER COUNT
02D5 20 4F 09   P          JSR     DOBUFC
02D8 A5 00      E          LDA     BUFSTR      ;RETURN A CHARACTER
02DA 85 00      E          STA     ROWCRS
02DC A5 01      E          LDA     BUFSTR+1
02DE 85 00      E          STA     COLCRS
02E0 A5 00      E  EGETC3  LDA     BUFCNT
02E2 F0 11                 BEQ     EGETC5
02E4 C6 00      E  EGETC7  DEC     BUFCNT      ;AND RETURN TILL BUFCNT=0
02E6 F0 0D                 BEQ     EGETC5
02E8 A5 00      E          LDA     DSTAT       ;IF ERR, LOOP ON EGETC7 UNTIL BUFR IS EMPTIE
02EA 30 F8                 BMI     EGETC7
02EC 20 04 02   P          JSR     GETCH
02EF 8D 00 00   E          STA     ATACHR
02F2 4C D0 0A   P          JMP     SWAP        ;AND RETURN WITHOUT TURNING CURSOR BACK ON
02F5 20 0B 07   P  EGETC5  JSR     DOCRWS      ;DO REAL CARRIAGE RETURN
02F8 A9 9B                 LDA     #CR         ;AND RETURN EOL
02FA 8D 00 00   E          STA     ATACHR
02FD 20 8F 02   P          JSR     RETUR2      ;TURN ON CURSOR THEN SWAP
0300 84 00      E          STY     DSTAT       ;SAVE KEYBOARD STATUS
0302 4C 00 0A   P          JMP     SWAP        ;AND RETURN THROUGH RETUR1
                ;
0305 6C 00 00   E  JBRIND  JMP     (ADRESS)    ;JSR TO THIS CAUSES JSR INDIRECT
                ;
0308 8D 00 00   E  EOUTCH  STA     ATACHR      ;SAVE ATASCII VALUE
030B 20 00 0A   P          JSR     SWAP
030E 20 66 07   P          JSR     ERANGE
0311 A9 00                 LDA     #0          ;CLEAR SUPER FUNCTION FLAG
0313 8D 00 00   E          STA     SUPERF
```

```
ERR LINE  ADDR  61 62 63 64      DISPLAY HANDLER  -- 10-30-78 --  DISPLC                    PAGE  11

    891  0320  20 C2 07     P  DOSS:   JSR    OFFCRS      ;TURN OFF CURSOR
    892  0323  36 DA 09     P          JSR    TSTCTL      ;TEST FOR CONTROL CHARACTERS (1=1 IF CTL)
    893  0326  F0 09                   BEQ    EOUTC5      ;CONTROL CODE
    894  0328  0E 00 00     E  EOUTC6: ASL    ESCFLG      ;ESCFLG ONLY WORKS ONCE
    895  032B  20 35 02     P          JSR    OUTCHE
    896  032E  4C 00 0A     F  ERETN:  JMP    SWAP        ;AND RETURN THROUGH RETUR1
    897  0331  AD 00 00     E  EOUTC5: LDA    DSPFLG      ;DO DSPFLG AND ESCFLG
    898  0334  0D 00 00     E          ORA    ESCFLG
    899  0337  D0 EF                   BNE    EOUTC6      ;IF NON-0 DISPLAY RATHER THAN EXECUTE IT
    900  0339  0E 00 00     E          ASL    ESCFLG
    901  033C  E8                      INX                ;PROCESS CONTROL CHARACTERS
    902  033D  AD 00 00     E          LDA    SUPERF      ;SUPER FUNCTION?
    903  0340  F0 05                   BEQ    EOUTC7      ;NO
    904  0342  8A                      TXA                ;YES -- ADJUST X REG
    905  0343  18                      CLC
    906  0344  69 20                   ADC    #SUPERS-CNTRLS-3
    907  0346  AA                      TAX
    908  0347  BD 53 0C     P  EOUTC7: LDA    CNTRLS,X    ;GET DISPLACEMENT INTO ROUTINE
    909  034A  85 00        E          STA    ADRESS
    910  034C  BD 54 0C     P          LDA    CNTRLS+1,X  ;GET HIGH BYTE
    911  034F  85 01        E          STA    ADRESS+1
    912  0351  20 0F 03     P          JSR    JSRIND      ;DO COMPUTED JSR
    913  0354  20 8F 02     P          JSR    RETUR2      ;DO CURSOR
    914  0357  4C 00 0A     P          JMP    SWAP        ;ALL DONE SO RETURN THROUGH RETUR1
    915                               ;
    916                               ; END SCREEN EDITOR
    917                               ;
```

```
                        PAGE
                ;
                ; BEGIN KEYBOARD HANDLER
                ;
0395 A9 FF      KGTOOO: LDA     #$FF
0397 8D 00 00   E       STA     CH
039A A9 00      KGETCH: LDA     #0              ;CLEAR SUPER FUNCTION FLAG
039C 8D 00 00   E       STA     SUPERF
039F A5 00      E       LDA     ICAX1Z          ;TEST LSB OF AUX1 FOR SPECIAL EDITOR READ MO
03A1 4A                 LSR     A
03A2 B0 7D              BCS     KGT075
03A4 A9 80              LDA     #BRKABT
03A6 AE 00 00   E       LDX     BRKKEY          ;TEST BREAK
03A9 F0 73              BEQ     KGT070          ;IF BREAK, PUT BRKABT IN DSTAT AND CR IN ATA
03AB AD 00 00   E       LDA     CH
03AE C9 FF              CMP     #$FF
03B0 F0 E9              BEQ     KGETCH
03B2 85 00      E       STA     HOLDCH          ;SAVE CH FOR SHIFT LOCK PROC
03B4 A2 FF              LDX     #$FF            ;"CLEAR" CH
03B6 8E 00 00   E       STX     CH
03B9 AE 00 00   E       LDX     NOCLIK          ;INHIBIT CLICK?
03BC D0 08              BNE     KGT002          ;YES.
03BE AE 00 00   E       LDX     KEYDIS          ;DISABLED KEYBOARD?
03C1 D0 03              BNE     KGT002          ;YES
03C3 20 21 0A   P       JSR     CLICK           ;DO KEYBOARD AUDIO FEEDBACK (A IS OK)
03C6 A8         KGT002: TAY                     ;DO ASCCON
03C7 C9 C0              CPY     #$C0            ;TEST FOR CTL & SHIFT TOGETHER
03C9 B0 C8              BCS     KGTOOO          ;CTRL-SHIFT COMBINATIONS IGNORED
                ;
03CB B1 00      E       LDA     (KEYDEF),Y
03CD 8D 00 00   E       STA     ATACHR          ;DONE
03D0 AE 00 00   E       LDX     KEYDIS          ;KEYBOARD DISABLED?
03D3 F0 04              BEQ     KGT008          ;NO.
                ;
                ; NOTE   KBD ENB/DIS MUST BE DEFINED IN 1ST LEVEL CONVERSION TABLE
                ;
03D5 C9 B6              CMP     #$B6            ;YES -- RECOGNIZE ONLY ENABLE FUNCTION
03D7 D0 B0              BNE     KGTOOO          ;IGNORE ALL ELSE
                ;
03D9 AA         KGT008  TAX                     ;SET CC
03DA 30 03              BMI     KGT010
03DC 4C 6B 04   P       JMP     KGTBOO          ;NOT SPECIAL CODE
                ;
03DF C9 80      KGT010  CMP     #$80            ;DO NULLS
03E1 F0 B3              BEQ     KGTOOO
                ;
03E3 C9 81              CMP     #$81            ;CHECK ATARI KEY
03E5 D0 0A              BNE     KGT020
                ;
03E7 AD 00 00   E       LDA     INVFLG
03EA 49 80              EOR     #$80
03EC 8D 00 00   E       STA     INVFLG
03EF B0 A5              BCS     KGTOOO          ;(JMP) DONT RETURN A VALUE
                ;
```

```
 972  0385 C9 82          KGT020  CMP   #$82      ;CAPS/LOWER
 973  0387 D0 0C                  BNE   KGT030
 974                      ;
 975  0389 AD 00 00     E         LDA   SHFLOK    ;LOWER CASE?
 976  038C F0 08                  BEQ   KGT040    ;YES -- GO TO CAPS LOCK
 977                      ;
 978  038E A9 00                  LDA   #0        ;CLEAR SHFLOK
 979  0390 8D 00 00     E         STA   SHFLOK
 980  0393 F0 75                  BEQ   KGT000    ;(JMP)
 981                      ;
 982  0395 C9 83          KGT030  CMP   #$83      ;SHIFT CAPS/LOWER
 983  0397 D0 07                  BNE   KGT050
 984                      ;
 985  0399 A9 40          KGT040  LDA   #$40
 986  039B 8D 00 00     E         STA   SHFLOK    ;SHIFT BIT
 987  039E D0 3A                  BNE   KGT000    ;(JMP).
 988                      ;
 989  03A0 C9 84          KGT050  CMP   #$84      ;CNTL CAPS/LOWER
 990  03A2 D0 08                  BNE   KGT060
 991                      ;
 992  03A4 A9 80                  LDA   #$80      ;CNTL BIT
 993  03A6 8D 00 00     E         STA   SHFLOK
 994  03A9 4C 5A 03     R         JMP   KGT000    ;(JMP)
 995                      ;
 996  03AC C9 85          KGT060  CMP   #$85      ;DO EOF
 997  03AE D0 0B                  BNE   KGT080
 998                      ;
 999  03E0 A9 88                  LDA   #EOFERR
1000  03E2 85 00        E KGT070  STA   DSTAT
1001  03E4 85 00        E         STA   BRKKEY    ;RESTORE BREAK
1002  03E6 A9 9B          KGT075  LDA   #CR       ;PUT CR IN ATACHR
1003  03E8 4C BE 04     R         JMP   KGT820    ;(UNCONDITIONAL)
1004                      ;
1005  03EB C9 86          KGT080  CMP   #$86      ;KEYBOARD ENABLE/DISABLE?
1006  03ED D0 12                  BNE   KGT090    ;NO
1007                      ;
1008  03EF AD 01 D3               LDA   PORTB     ;TOGGLE LED 1
1009  03F2 49 04                  EOR   #$04
1010  03F4 8D 01 D3               STA   PORTB
1011                      ;
1012  03F7 AD 00 00     E         LDA   KEYDIS    ;YES -- TOGGLE THE VARIABLE
1013  03FA 49 FF                  EOR   #$FF
1014  03FC 8D 00 00     E         STA   KEYDIS
1015  03FF B0 13                  BCS   KGT112    ;(JMP) TO COMMON CODE
1016                      ;
1017                      ;
1018  0401 C9 88          KGT090  CMP   #$88      ;GONZO FUNCTION?
1019  0403 D0 03                  BNE   KGT110    ;NO
1020                      ;
1021  0405 4C 05 04     R         JMP   *         ;YES.
1022                      ;
1023  0408 C9 89          KGT110  CMP   #$89      ;KEY CLICK ON-OFF?
1024  040A D0 10                  BNE   KGT120    ;NO.
1025                      ;
```

```
 040C  AD 00 00   E            LDA    NOCLIK       ;YES -- TOGGLE THE VARIABLE
 040F  49 FF                   EOR    #$FF
 0411  8D 00 00   E            STA    NOCLIK
 0414  D0 03        KGT110:    BNE    KGT115
                    ;
 0416  20 21 0A   P            JSR    CLICK        ;PRODUCE MISSING CLICK
                    ;
 0419  4C 5A 03   P KGT115:    JMP    KGT000       ;*** SKIP BRANCH POINT ***
                    ;
 041C  C9 BE        KGT120:    CMP    #$BE         ;CODE < $BE?
 041E  B0 12                   BCS    KGT130       ;NO
 0420  C9 BA                   CMP    #$BA         ;CODE >= BA?
 0422  90 F5                   BCC    KGT115       ;NO -- IGNORE.
 0424  E9 BA                   SBC    #$BA         ;NORMALIZE $BA-BD TO 0-3
 0426  06 00      E            ASL    HOLDCH       ;TEST FOR SHIFT KEY
 0428  10 02                   BPL    KGT125       ;NO SHIFT KEY
                    ;
 042A  09 04                   ORA    #4           ;SHIFT-FX -- INDEX = 4-7
                    ;
 042C  A8           KGT125:    TAY                 ;INDEX TO FUNCTION KEY DEFS
 042D  B1 00      E            LDA    (FKDEF),Y
 042F  4C 91 03   P            JMP    KGT007       ;TRY AGAIN.
                    ;
 0432  C9 92        KGT130:    CMP    #$92         ;CODE < $92?
 0434  B0 0D                   BCS    KGT170       ;NO
 0436  C9 8E                   CMP    #$8E         ;CODE >= $8E?
 0438  90 DF                   BCC    KGT115       ;NO -- IGNORE.
 043A  E9 72                   SBC    #$8E-$1C     ;CONVERT $8E-91 TO $1C-1F
 043C  EE 00 00   E            INC    SUPERF       ;SET SUPER FUNCTION FLAG
 043F  D0 40                   BNE    KGT820       ;(JMP)
                    ;
 0441  C9 92        KGT170:    CMP    #$92         ;SWITCH CHARACTER SETS?
 0443  D0 23                   BNE    KGT180       ;NO
                    ;
 0445  AD 00 00   E            LDA    CHBAS        ;INTERNATIONAL SET SELECTED?
 0448  29 FC                   AND    #$FC
 044A  C9 CC                   CMP    #INTCHR
 044C  F0 0A                   BEQ    KGT175       ;YES -- SWITCH TO DOMESTIC.
                    ;
 044E  A2 CC                   LDX    #INTCHR      ;NO -- SELECT INTERNATIONAL.
 0450  AD 01 D3                LDA    PORTB        ;LIGHT LED 2.
 0453  29 F7                   AND    #$F7
 0455  4C 5F 04   P            JMP    KGT177
                    ;
 0458  A2 E0        KGT175:    LDX    #DOMCHR      ;SELECT DOMESTIC CHARSET.
 045A  AD 01 D3                LDA    PORTB        ;UNLIGHT LED 2.
 045D  09 08                   ORA    #$08
                    ;
 045F  8D 01 D3     KGT177:    STA    PORTB
 0462  8E 00 00   E            STX    CHBAS
 0465  4C 5A 03   P            JMP    KGT000
                    ;
 0468             P KGT180:    =      *
 0468  A5 00      E KGT800:    LDA    HOLDCH       ;PROCESS SHIFT LOCKS
```

```
1080 048A C9 40          CMP   #$40      ;REGULAR SHIFT AND CONTROL TAKE PRECEDENCE
1081 048C B0 15          BCS   KGT810    ;OVER LOCK
1082 048E AD 00 00    E  LDA   ATACHR    ;TEST FOR ALPHA
1083 0491 C9 61          CMP   #$61      ;LOWER CASE A
1084 0493 90 0E          BCC   KGT810    ;NOT ALPHA IF LT
1085 0495 C9 7B          CMP   #$7B      ;LOWER CASE Z+1
1086 0497 B0 0A          BCS   KGT810    ;NOT ALPHA IF GE
1087 0499 AD 00 00    E  LDA   SHFLOK    ;DO SHIFT/CONTROL LOCK
1088 049C F0 05          BEQ   KGT810    ;IF NO LOCK, DONT RE-DO IT
1089 049E 05 00       E  ORA   HOLDCH
1090 04A0 4C BA 03    P  JMP   KGT002    ;DO RETRY
1091 04A3 20 DA 09    P  KGT810: JSR   TSTCTL    ;DONT INVERT MSB OF CONTROL CHARACTERS
1092 04A6 F0 09          BEQ   KGT830
1093 04A8 AD 00 00    E  LDA   ATACHR
1094 04AB 4D 00 00    E  EOR   INVFLG
1095 04AE 8D 00 00    E  KGT820: STA   ATACHR
1096 04B1 4C A2 02    P  KGT830: JMP   RETUR1    ALL DONE
1097                     ;
1098                     ; END KEYBOARD HANDLER
1099                     ;
```

```
1100                                    PAGE
1101
1102                            ; CONTROL CHARACTER PROCESSORS
1103                            ;
1104                            ;
1105                            ; ESCAPE
1106                            ;
1107  0494  A9 80               ESCAPE: LDA     #$80        ;SET ESCAPE FLAG
1108  0496  8D 00 00         E         STA     ESCFLG
1109  0499  60                       RTS
1110                            ;
1111                            ; CURSOR UP
1112                            ;
1113  049A  C6 00             E CRSRUP  DEC     ROWCRS
1114  049C  10 06                    BPL     COMRET
1115  049E  AE 00 00          E         LDX     BOTSCR      ;WRAPAROUND
1116  04A1  CA                        DEX
1117  04A2  86 00             E UPDNCM: STX     ROWCRS
1118  04A4  4C AA 09          P COMRET  JMP     STRBEG      ;CONVERT ROW AND COL TO LOGCOL AND RETURN
1119
1120                            ; CURSOR DOWN
1121                            ;
1122  04A7  E6 00             E CRSRDN  INC     ROWCRS
1123  04A9  A5 00             E         LDA     ROWCRS
1124  04AB  CD 00 00          E         CMP     BOTSCR
1125  04AE  90 F4                     BCC     COMRET
1126  04B0  A2 00                     LDX     #0
1127  04B2  F0 EE                     BEQ     UPDNCM      ;(UNCONDITIONAL)
1128                            ;
1129                            ; CURSOR LEFT
1130                            ;
1131  04B4  C6 00             E CRSRLF: DEC     COLCRS
1132  04B6  A5 00             E         LDA     COLCRS
1133  04B8  30 04                     BMI     CRSRL1      ;(IF LMARGN=0, THIS ELIMINATES PROBLEM CASE)
1134  04BA  C5 00             E         CMP     LMARGN
1135  04BC  B0 04                     BCS     COMRE1
1136  04BE  A5 00             E CRSRL1: LDA     RMARGN
1137  04C0  85 00             E LFRTCM: STA     COLCRS
1138  04C2  4C 2C 09          P COMRE1  JMP     DOLCOL      ;CONVERT ROW AND COL TO LOGCOL AND RETURN
1139
1140                            ; CURSOR RIGHT
1141                            ;
1142  04C5  E6 00             E CRSRRT: INC     COLCRS
1143  04C7  A5 00             E         LDA     COLCRS
1144  04C9  C5 00             E         CMP     RMARGN
1145  04CB  90 F5                     BCC     COMRE1
1146  04CD  F0 F3                     BEQ     COMRE1      ;(CAUSE BLE)
1147  04CF  A5 00             E CRSRR1: LDA     LMARGN
1148  04D1  4C C0 D4          P         JMP     LFRTCM      ;UNCONDITIONAL TO COMMON STORE
1149                            ;
1150                            ; CLEAR SCREEN
1151                            ;
1152  04D4  20 44 DA          P CLRSCR: JSR     PUTMSC
1153  04D7  A4 00             E         LDY     ADRESS
```

```
LOC CMD  ADDR  01 02 03 04      DISPLAY LABELS  -- 10-30-79 --  DISPLC                PAGE  17

 1154  04B5  A9 00              LDA   #0
 1155  04B7  85 00          E   STA   ADRESS
 1156  04B9  85 00          E   CLRSC2  STA   (ADRESS),Y  ;AC IS ZERO!
 1157  04BB  C8                  INY
 1158  04BC  D0 FB              BNE   CLRSC2
 1159  04BE  E6 01          E   INC   ADRESS+1
 1160  04C0  A6 01          E   LDX   ADRESS+1
 1161  04C2  E4 CD          E   CPX   RAMTOP
 1162  04C4  90 F2              BCC   CLRSC2
 1163  04C6  A9 FF              LDA   #$FF        ;CLEAR UP LOGICAL LINE BIT MAP
 1164  04C8  99 00 00      E   CLRSC3  STA   LOGMAP,Y  ;(Y IS ZERO AFTER CLRSC2 LOOP)
 1165  04CB  C8                  INY
 1166  04CC  C0 04              CPY   #4
 1167  04CE  90 F8              BCC   CLRSC3
 1168
 1169                          ; CURSOR HOME
 1170                          ;
 1171  04D0  20 28 DA      P   HOME    JSR   COLCR       ;PLACE COLCRS AT LEFT EDGE
 1172  04D3  85 00          E           STA   LOGCOL
 1173  04D5  85 03          E           STA   BUFSTR+1
 1174  04D7  A9 00                      LDA   #0
 1175  04D9  85 00          E           STA   ROWCRS
 1176  04DB  85 01          E           STA   COLCRS+1
 1177  0501  85 00          E           STA   BUFSTR
 1178  0502  60                       RTS
 1179
 1180                          ; BACKSPACE
 1181                          ;
 1182  050A  A5 00          E   BB      LDA   LOGCOL      ;BACKSPACE
 1183  050A  C5 00          E           CMP   LMARGN
 1184  050B  F0 21                      BEQ   BS1
 1185  050A  A5 00          E   BBA     LDA   COLCRS      ;LEFT EDGE?
 1186  050C  C5 00          E           CMP   LMARGN
 1187  050E  D0 03                      BNE   BS3         ;NO
 1188  0510  20 C1 05      P           JSR   DELTIM      ;YES: SEE IF LINE SHOULD BE DELETED
 1189  0513  20 84 04      P   BS3     JSR   CRSRLF
 1190  0516  A5 00          E           LDA   COLCRS
 1191  0518  C5 00          E           CMP   RMARGN
 1192  051A  00 07                      BNE   BS2
 1193  051C  A3 00          E           LDA   ROWCRS
 1194  051E  F0 03                      BEQ   BS2
 1195  0520  20 9A 04      P           JSR   CRSRUP
 1196  0523  A9 20                  BS2 LDA   #$20        ;MAKE BACKSPACE DESTRUCTIVE
 1197  0525  8D 00 00      E           STA   ATACHR
 1198  0528  20 4E 02      P           JSR   OUTPUT
 1199  0528  4C 2C 09      P   BS1:    JMP   DOLCOL      ;AND RETURN
 1200                          ;
 1201                          ; TAB CURSOR
 1202                          ;
 1203  052E  20 C5 04      P   TAB     JSR   CRSRRT      ;BEGIN SEARCH
 1204  0531  A5 00          E           LDA   COLCRS      ;TEST FOR NEW LINE
 1205  0533  C5 00          E           CMP   LMARGN
 1206  0535  D0 09                      BNE   TAB1        ;NO
 1207  0537  20 0F 07      P           JSR   DOCR        ;DO CARRIAGE RETURN
```

```
 1208  0528  20 0A 08     P          JSR     LOGSET      ;CHECK IF END OF LOGICAL LINE
 1209  052B  B0 07        P          BCS     TAB2        ;YES
 1210  052D  A5 00        E  TAB1:   LDA     LOGCOL      ;CHECK FOR TAB STOP
 1211  052F  20 07 08     P          JSR     BITSET
 1212  0532  90 E8        P          BCC     TAB         ;NO, SO KEEP LOOKING
 1213  0534  4C 2C 09     P  TAB2:   JMP     DOLCOL      ;CONVERT ROW AND COL TO LOGCOL AND RETURN
 1214
 1215                             ; SET TAB
 1216                             ;
 1217  0537  A5 00        E  SETTAB: LDA     LOGCOL
 1218  0539  4C E6 07     P          JMP     BITSET      ;SET BIT IN MAP AND RETURN
 1219
 1220                             ; CLEAR TAB
 1221                             ;
 1222  053E  A5 00        E  CLRTAB: LDA     LOGCOL
 1223  0540  4C F4 07     P          JMP     BITCLR      ;CLEAR " " " " "
 1224
 1225                             ; INSERT CHARACTER
 1226                             ;
 1227  0543  20 EA 09     P  INSCHR: JSR     PHACRS
 1228  0546  20 13 02     P          JSR     GETPLT      ;GET CHARACTER UNDER CURSOR
 1229  0549  85 00        E          STA     INSDAT
 1230  054B  A9 00                   LDA     #0
 1231  054D  8D 00 00     E          STA     SCRFLG
 1232  0550  20 60 02     P  INSCH4: JSR     OUTCH2      ;STORE DATA
 1233  0553  A5 00        E          LDA     LOGCOL      ;SAVE LOGCOL.  IF AFTER INCRSA LOGCOL IS
 1234  0555  48                      PHA                 ;< THAN IT IS NOW, END LOOP
 1235  0556  20 BC 06     P          JSR     INCRSA      ;SPECIAL INCRSR ENTRY POINT
 1236  0559  68                      PLA
 1237  055A  C5 00        E          CMP     LOGCOL
 1238  055C  B0 0C        E          BCS     INSCH3      ;QUIT
 1239  055E  A5 00        E  INSCH1: LDA     INSDAT      ;KEEP GOING
 1240  0560  48                      PHA
 1241  0561  20 13 02     P          JSR     GETPLT
 1242  0564  85 00        E          STA     INSDAT
 1243  0566  68                      PLA
 1244  0567  4C 60 05     P          JMP     INSCH4
 1245  056A  20 F5 09     P  INSCH3: JSR     PLACRS
 1246  056D  CE 00 00     E  INSCH6: DEC     SCRFLG
 1247  0570  30 04        E          BMI     INSCH5      ;IF SCROLL OCCURRED
 1248  0572  C6 00        E          DEC     ROWCRS      ;MOVE CURSOR UP
 1249  0574  D0 F7                   BNE     INSCH6      ;(UNCOND) CONTINUE UNTIL SCRFLG IS MINUS
 1250  0576  4C 2C 09     P  INSCH5: JMP     DOLCOL      ;CONVERT ROW AND COL TO LOGCOL AND RETURN
 1251
 1252                             ; DELETE CHARACTER
 1253                             ;
 1254  0579  20 EA 09     P  DELCHR: JSR     PHACRS
 1255  057C  20 56 06     P  DELCH1: JSR     CONVRT      ;GET DATA TO THE RIGHT OF THE CURSOR
 1256  057F  A5 00        E          LDA     ADRESS
 1257  0581  85 00        E          STA     SAVADR      ;SAVE ADRESS TO KNOW WHERE TO PUT DATA
 1258  0583  A5 01        E          LDA     ADRESS+1
 1259  0585  85 01        E          STA     SAVADR+1
 1260  0587  A5 00        E          LDA     LOGCOL
 1261  0589  48                      PHA
```

```
1262 059A 20 B4 06    P          JSR     INCRSR      ;PUT CURSOR OVER NEXT CHARACTER
1263 059D 68                     PLA
1264 059E C5 00       E          CMP     LOGCOL      ;TEST NEW LOGCOL AGAINST OLD LOGCOL
1265 05A0 B0 10                  BCS     DELCH2      ;IF OLD GE.NEW THEN QUIT
1266 05A2 A5 00       E          LDA     ROWCRS      ;IS ROW OFF SCREEN?
1267 05A4 CD 00 00    E          CMP     BOTSCR
1268 05A7 B0 09                  BCS     DELCH2      ;YES, SO QUIT
1269 05A9 20 13 02    P          JSR     GETPLT      ;GET DATA UNDER CURSOR
1270 05AC A0 00                  LDY     #0
1271 05AE 91 00       E          STA     (SAVADR),Y  ;PUT IT IN PREVIOUS POSITION
1272 05B0 F0 DA                  BEQ     DELCH1      ;AND LOOP (UNCONDITIONAL)
1273 05B2 A0 00       DELCH2:    LDY     #0
1274 05B4 98                     TYA
1275 05B5 91 00       E          STA     (SAVADR),Y  ;CLEAR THE LAST POSITION
1276 05B7 20 B6 09    P          JSR     DELTIA      ;TRY TO DELETE A LINE
1277 05BA 20 F5 09    P          JSR     PLACRS
1278 05BD 4C 2C 09    P          JMP     DOLCOL      ;AND RETURN
1279                             ;
1280                             ;  INSERT LINE
1281                             ;
1282 05C0 38          INSLIN:    SEC                 ;NORMAL INSLIN PUTS "1" INTO BIT MAP
1283 05C1 20 6C 08    P  INSLIA: JSR     EXTEND      ;ENTRY POINT FOR C=0
1284 05C4 A5 00       E          LDA     LMARGN      ;DO CARRIAGE RETURN (NO LF)
1285 05C6 85 00       E          STA     COLCRS
1286 05C8 20 56 06    P          JSR     CONVRT      ;GET ADDRESS
1287 05CB 20 38 08    P          JSR     MOVLIN
1288 05CE 20 BC 08    P          JSR     CLRLIN      ;CLEAR CURRENT LINE
1289 05D1 4C 2C 09    P          JMP     DOLCOL      ;CONVERT ROW AND COL TO LOGCOL AND RETURN
1290                             ;
1291                             ;  DELETE LINE
1292                             ;
1293 05D4 20 2C 09    P  DELLIN: JSR     DOLCOL      ;GET BEGINNING OF LOG LINE (HOLD1)
1294 05D7 A4 00       E  DELLIA: LDY     HOLD1       ;SQUEEZE BIT MAP
1295 05D9 84 00       E          STY     ROWCRS      ;PUT CURSOR THERE
1296 05DB A4 00       E  DELLIB: LDY     ROWCRS
1297 05DD 98          DELLI1:    TYA
1298 05DE 38                     SEC
1299 05DF 20 05 08    P          JSR     LO2GET      ;GET NEXT BIT
1300 05E2 08                     PHP
1301 05E3 98                     TYA
1302 05E4 18                     CLC
1303 05E5 69 78                  ADC     #120
1304 05E7 28                     PLP
1305 05E8 20 E6 07    P          JSR     BITPUT      ;WRITE IT OVER PRESENT BIT
1306 05EB C8                     INY
1307 05EC C0 18                  CPY     #24
1308 05EE D0 E9                  BNE     DELLI1      ;LOOP
1309 05F0 AD 02 00    E          LDA     LOGMAP+2    ;SET LSB
1310 05F3 09 01                  ORA     #1
1311 05F5 8D 02 00    E          STA     LOGMAP+2
1312 05F8 A5 00       E  DELLI2: LDA     LMARGN      ;DELETE LINE OF DATA USING PART OF SCROLL
1313 05FA 85 00       E          STA     COLCRS      ;CR, NO LF
1314 05FC 20 56 06    P          JSR     CONVRT
1315 05FF 20 D4 0B    P          JSR     SCROL1
```

```
1316  0602  20 02 09      P          JSR     LOGGET      ;TEST NEXT LINE FOR CONTINUATION
1317                             ; IS IT A NEW LOG LINE?
1318  0605  90 04                    BCC     DELL18      ;NO SO DELETE ANOTHER
1319  0607  4C 2C 09      P         ~~JMP     DOLCOL~~    ;YES SO DOLCOL AND RETURN
                                    BCS     CHLEFT
1320                             ;
1321                             ; BELL
1322                             ;
1323  060A  A0 20               BELL:   LDY     #$20
1324  060C  20 21 0A      P    BELL1:  JSR     CLICK
1325  060F  88                       DEY
1326  0610  10 FA                    BPL     BELL1
1327  0612  60                       RTS
1328                             ;
1329                             ; SUPER FUNCTIONS
1330                             ;
1331  0613  4C F4 04      P    CHOME:  JMP     HOME        ;CURSOR HOME
1332                             ;
1333  0616  20 F4 04      P    CHBOT:  JSR     HOME        ;CURSOR TO BOTTOM OF SCREEN
1334  0619  4C 9A 04      P            JMP     CRSRUP
1335                             ;
1336  061C  4C CF 04      P    CHLEFT: JMP     CRSRR1      ;CURSOR TO LEFT MARGIN
1337                             ;
1338  061F  4C BE 04      P    CHRIGT: JMP     CRSRL1      ;CURSOR TO RIGHT MARGIN
```

```
1339                        PAGE
1340
1341
1342                    ; ROUTINES
1343                    ;
1344                    ;
1345                    ; DOUBLE BYTE DECREMENT OF INDIRECT POINTER
1346                    ; INCLUDING OR SUBTRACT AND DB DOUBLE DECREMENT
1347                    ;
1348  0622  A9 02       DBDDEC  LDA     #2
1349  0624  D0 0A               BNE     DBSUB           ;(UNCONDITIONAL)
1350                    ;
1351                    ; STORE DATA INDIRECT AND DECREMENT POINTER
1352                    ; (PLACED HERE TO SAVE JMP DBDEC AFTER STORE)
1353  0626  A4 00       E STORE   LDY     DSTAT           ;RETURN ON ERROR
1354  0628  30 28               BMI     STROK
1355  062A  A0 00               LDY     #0
1356  062C  91 00       E STORE1  STA     (ADRESS),Y
1357                    ;         JMP     DBDEC           DECREMENT AND RETURN
1358                    ;
1359  062E  A9 01       DBDEC   LDA     #1
1360  0630  8D 00 00    E DBSUB   STA     SUBTMP
1361  0633  A3 00       E         LDA     DSTAT           ;RETURN ON ERROR
1362  0635  30 1E               BMI     STROK
1363  0637  A5 00       E         LDA     ADRESS
1364  0659  38                  SEC
1365  063A  ED 00 00    E         SBC     SUBTMP
1366  063D  85 00       E         STA     ADRESS
1367  063F  B0 02               BCS     DBSUB1
1368  0641  C6 01       E         DEC     ADRESS+1
1369  0643  A5 01       E DBSUB1  LDA     APPMHI+1        ;MAKE SURE NOTHING EVER OVERWRITES APPMHI
1370  0645  C5 01       E         CMP     ADRESS+1
1371  0647  90 0C               BCC     STROK           ;OK
1372  0649  D0 06               BNE     STRERR          ;ERROR
1373  064B  A5 00       E         LDA     APPMHI
1374  064D  C5 00       E         CMP     ADRESS
1375  064F  90 04               BCC     STROK
1376  0651  A9 93       STRERR  LDA     #SCRMEM         ;SHOW MEM TOO SMALL FOR SCREEN ERROR
1377  0653  25 00       E         STA     DSTAT
1378  0655  60          STROK   RTS
1379                    ;
1380
1381                    ;
1382                    ; CONVERT ROW/COLUMN CURSOR INTO REAL ADDRESS (FROM SAVMSC ON UP)
1383                    ;
1384  0656  A2 01       CONVRT  LDX     #01
1385  0658  8A 00       E         STX     MLTTMP          ;VERTICAL CALCULATIONS
1386  065A  CA                  DEX                     ;VERTICAL CALCULATIONS
1387  065B  86 01       E         STX     ADRESS+1        ;CLEAR HI BYTE
1388  065D  A5 00       E         LDA     ROWCRS          ;ADRESS = ROWCRS*5
1389  065F  0A                  ASL     A               ;MULTIPLY BY 4
1390  0660  26 01       E         ROL     ADRESS+1
1391  0662  0A                  ASL     A
1392  0663  26 01       E         ROL     ADRESS+1        ;CLEAR CARRY
```

```
1350  0663  65 00     E          ADC     ROWCRS          ;ADD TO MAKE Y*8
1394  0667  85 00     E          STA     ADRESS
1395  0669  90 02                BCC     CONVR0
1396  066B  E6 01     E          INC     ADRESS+1
1397  066D  A4 00     E  CONVR0  LDY     DINDEX          ;GET MODE
1398  066F  BE F7 08  P          LDX     DMLINE,Y        ;GET NUMBER OF SHIFTS
1399  0672  06 00     E  CONVR1  ASL     ADRESS          ;ADRESS := ADRESS *X
1400  0674  26 01     E          ROL     ADRESS+1        ;DO THE DIVIDE
1401  0676  CA                  DEX
1402  0677  D0 F9                BNE     CONVR1
1403  0679  A5 01     E          LDA     COLCRS+1        ;HORIZONTAL CALCULATIONS.
1404  067B  4A                  LSR     A               ;SAVE LSB FOR LATER
1405  067C  A5 00     E          LDA     COLCRS          ;GET LOW BYTE
1406  067E  BE 2A 0C  P          LDX     DIV2TB,Y        ;GET SHIFT AMOUNT.
1407  0681  F0 06                BEQ     CONVR3          ;CARRY CLEAR IF NO SHIFT
1408  0683  6A         CONVR2:  ROR     A               ;ROLL IN THE CARRY.
1409  0684  06 00     E          ASL     MLTTMP          ;SHIFT INDEX
1410  0686  CA                  DEX
1411  0687  D0 FA                BNE     CONVR2
1412  0689  65 00     E  CONVR3: ADC     ADRESS          ;CARRY IS ALWAYS CLEAR.
1413  068B  90 02                BCC     CONVR4
1414  068D  E6 01     E          INC     ADRESS+1
1415  068F  18         CONVR4:  CLC
1416  0690  65 00     E          ADC     SAVMSC
1417  0692  85 00     E          STA     ADRESS
1418  0694  85 00     E          STA     OLDADR
1419  0696  A5 01     E          LDA     ADRESS+1
1420  0698  65 01     E          ADC     SAVMSC+1
1421  069A  85 01     E          STA     ADRESS+1
1422  069C  85 01     E          STA     OLDADR+1
1423  069E  BE 2A 0C  P          LDX     DIV2TB,Y
1424  06A1  BD 4A 0C  P          LDA     HMASK,X
1425  06A4  25 00     E          AND     COLCRS
1426  06A6  63 00     E          ADC     MLTTMP
1427  06AB  A8                   TAY                     ;MAKE A NEW INDEX.
1428  06A9  B9 3A 0C  P          LDA     DMASKT-1,Y      ;GET THE FINAL MASK.
1429  06AC  8D 00 00  E          STA     DMASK
1430  06AF  85 00     E          STA     SHFAMT
1431  0681  A0 00                LDY     #00             ;SET Y TO ZERO.
1432  0683  60         INCRET:  RTS
1433                              ;
1434                              ;
1435                              ; INCREMENT CURSOR AND DETECT BOTH END OF LINE AND END OF SCREEN
1436                              ;
1437  0684  A9 00      INCRSB:  LDA     #0              ;NON-EXTEND ENTRY POINT
1438  0686  F0 02                BEQ     INCRSC
1439  0688  A9 98      INCRSR:  LDA     #$98            ;SPECIAL CASE ELIMINATOR
1440  068A  85 00     E  INCRSC  STA     INSDAT
1441  068C  E6 00     E  INCRSA  INC     LOGCOL          ;(INSCHR ENTRY POINT)
1442  068E  E6 00     E          INC     COLCRS
1443  0690  D0 02                BNE     INCRS2          ;DO HIGH BYTE
1444  0692  E6 01     E          INC     COLCRS+1
1445  0694  A5 00     E  INCRS2  LDA     COLCRS          ;TEST END OF LINE
1446  0696  A6 00     E          LDX     DINDEX
```

```
1447 06CB DD 08 0C   R        CMP   COLUMN,X   ;TEST TABLED VALUE FOR ALL SCREEN MODES
1448 06CE F0 0A              BEQ   INC2A      ;DO CR IF EQUAL
1449 06D0 E0 00              CPX   #0         ;MODE 0?
1450 06D2 D0 E2              BNE   INCRET     ;IF NOT, JUST RETURN
1451 06D4 C5 00         E    CMP   RMARGN     ;TEST AGAINST RMARGN
1452 06D6 F0 0E              BEQ   INCRET     ;EQUAL IS OK
1453 06D8 90 0C              BCC   INCRET     ;LESS THAN IS O.K.
1454 06DA E0 08         INC2A: CPX   #8         ;CHECK MODE
1455 06DC D0 04              BNE   DOCR1      ;NOT 320x1 SO DO IT
1456 06DE A5 01         E    LDA   COLCRS+1   ;TEST MSB
1457 06E0 F0 04              BEQ   INCRET     ;ONLY AT 64 SO DON'T DO IT
1458 06E2 A5 00         E    DOCR1 LDA   DINDEX     ;DON'T MESS WITH LOGMAP IF NO MODE ZERO
1459 06E4 D0 2C              BNE   DOCR
1460 06E6 A5 00         E    LDA   LOGCOL     ;TEST LINE OVERRUN
1461 06E8 C9 81              CMP   #81
1462 06EA 90 0A              BCC   DOCR1B     ;IF LESS THAN 81 IT IS DEFINITELY NOT LINE 3
1463 06EC A5 00         E    LDA   INSDAT
1464 06EE F0 22              BEQ   DOCR       ;ONLY DO LOG LINE OVERFLOW IF INSDAT <>0
1465 06F0 20 0B 07      P    JSR   DOCRWS     ;LOG LINE OVERFLOW IS SPECIAL CASE
1466 06F3 4C 35 07      P    JMP   INCRS1     ;RETURN
1467 06F6 20 0F 07      P    DOCR1B JSR  DOCR     ;GET IT OVER WITH
1468 06F9 A5 00         E    LDA   ROWCRS
1469 06FB 18                 CLC              ;TEST LOGICAL LINE BIT MAP
1470 06FC 69 78              ADC   #120
1471 06FE 20 D7 08      P    JSR   BITGET
1472 0701 90 08              BCC   DOCR1A     ;DON'T EXTEND IF OVERRUN IS INTO MIDDLE OF L
1473 0703 A5 00         E    LDA   INSDAT     ;DON'T EXTEND IF INSDAT IS ZERO
1474 0705 F0 04              BEQ   DOCR1A     ;(INSCHR SPECIAL CASE)
1475 0707 18                 CLC              ;INSERT "0" INTO BIT MAP
1476 0708 20 C1 05      P    JSR   INBLIA
1477 070B 4C 2C 09      P    DOCR1A JMP  DOLCOL     ;CONVERT ROW AND COL TO LOGCOL AND RETURN
1478 070E A9 9B         P    DOCRWS LDA  #$9B     ;DOCR WITH SCROLLING (NORMAL MODE)
1479 0710 85 00         E    STA   INSDAT
1480 0712 20 35 0A      P    DOCR: JSR   COLCR    ;PLACE COLCRS AT LEFT EDGE
1481 0715 A9 00              LDA   #0
1482 0717 85 01         E    STA   COLCRS+1
1483 0719 E6 00         E    INC   ROWCRS
1484 071B A6 00         E    DOCR2: LDX  DINDEX
1485 071D A0 18              LDY   #24        ;SET UP SCROLL LOOP COUNTER
1486 071F 24 00         E    BIT   SWPFLG
1487 0721 10 05              BPL   DOCR2A     ;BRANCH IF NORMAL
1488 0723 A0 04              LDY   #4
1489 0725 98                 TYA
1490 0726 D0 03              BNE   DOCR2B     ;(UNCONDITIONAL)
1491 0728 BD 1F 0C      P    DOCR2A LDA  NDROWS,X ;GET NO OF ROWS
1492 072B C6 00         E    DOCR2B CMP  ROWCRS
1493 072D D0 29              BNE   INCRS1
1494 072F 8C 00 00      E    STY   HOLD3
1495 0732 8A                 TXA              ;DON'T SCROLL IF MODE <> 0
1496 0733 D0 23              BNE   INCRS1
1497 0735 A5 00         E    LDA   INSDAT     ;OR IF INSDAT = 0
1498 0737 F0 1F              BEQ   INCRS1
1499                         ;    LDA   INSDAT     IF INSDAT <> $9B THEN ROLL IN A 0
1500 0739 C9 9B              CMP   #$9B       ;TO EXTEND BOTTOM LOGICAL LINE
```

```
1501                            SEC
1502  0736  F0 01                BEQ     DOCR4B          ;(CARRY WILL BE SET)
1503  0738  18                   CLC
1504  0739  20 A1 1B      P  DOCR4A  JSR     SCROLL          ;LOOP BACK TO HERE IF >1 SCROLLS
1505  073C  EE 00 00      E          INC     SCRFLG
1506  073F  CA 00         E          DEC     BUFSTR          ;ROWS MOVE UP SO BUFSTR SHOULD TOO
1507  0741  10 02                   BPL     DOCR4D
1508  0743  E6 00         E          INC     BUFSTR
1509  0745  CE 00 00      E  DOCR4D  DEC     HOLD3
1510  0748  AD 00 00      E          LDA     LOGMAP
1511  074B  38                      SEC                     ;FOR PARTIAL LINES, ROLL IN A "1"
1512  074C  10 E8                   BPL     DOCR4B          ;AGAIN IF PARTIAL LOGICAL LINE
1513  074E  AD 00 00      E          LDA     HOLD3           ;PLACE CURSOR AT NEW LINE NEAR THE BOTTOM
1514  0751  85 00         E          STA     ROWCRS
1515  0753  4C 2C 09      P  INCRS1  JMP     DOLCOL          ;CONVERT ROW AND COL TO LOGCOL AND RETURN
1516                            ;
1517                            ;
1518                            ; SUBEND  SUBTRACT ENDPT FROM ROWAC OR COLAC  (X=0 OR 2)
1519                            ;
1520  0756  38            SUBEND  SEC
1521  0757  B5 00         E          LDA     ROWAC,X
1522  0759  E5 00         E          SBC     ENDPT
1523  075B  95 00         E          STA     ROWAC,X
1524  075D  B5 01         E          LDA     ROWAC+1,X
1525  075F  E5 01         E          SBC     ENDPT+1
1526  0761  95 01         E          STA     ROWAC+1,X
1527  0763  60                       RTS
1528                            ;
1529                            ;
1530                            ; RANGE:  DO CURSOR RANGE TEST.  IF ERROR, POP STACK TWICE AND JMP RETURN
1531                            ;         (ERANGE IS EDITOR ENTRY POINT AND TEST IF EDITOR IS OPEN.
1532                            ;          IF IT ISNT IT OPENS THE EDITOR AND CONTINUES)
1533                            ;
1534  0766  AD 00 00      E  ERANGE* LDA     BOTSCR          ;IF BOTSCR=4
1535  0769  C9 04                   CMP     #4
1536  076B  F0 07                   BEQ     RANGE           ;THEN IT IS IN MIXED MODE AND OK
1537  076D  A5 00         E          LDA     DINDEX          ;IF MODE = 0
1538  076F  F0 03                   BEQ     RANGE           ;THEN IT IS IN EDITOR MODE AND OK
1539  0771  20 37 00      P          JSR     EOPEN           ;IF NOT, OPEN EDITOR
1540  0774  A9 27         RANGE:     LDA     #39             ;***** RANGE CHECK RMARGN ***** SET UP AC
1541  0776  C5 00         E          CMP     RMARGN          ;***** RANGE CHECK RMARGN ***** COMPARE
1542  0778  B0 02                   BCS     RANGE3          ;***** RANGE CHECK RMARGN ***** BRANCH GE
1543  077A  85 00         E          STA     RMARGN          ;***** RANGE CHECK RMARGN ***** BAD SO STORE
1544  077C  A6 00         E  RANGE3  LDX     DINDEX
1545  077E  B0 19 DC      P          LDA     NOROWS,X        ;CHECK ROWS
1546  0781  C5 00         E          CMP     ROWCRS
1547  0783  90 2A                   BCC     RNGERR          ;(ERROR IF TABLE GE ROWCRS)
1548  0785  F0 28                   BEQ     RNGERR
1549  0787  E0 08                   CPX     #8              ;CHECK FOR 320X1
1550  0789  D0 0A                   BNE     RANGE1          ;SPECIAL CASE IT
1551  078B  A5 01         E          LDA     COLCRS+1
1552  078D  F0 13                   BEQ     RNGOK           ;IF HIGH BYTE IS 0, COL IS OK
1553  078F  C9 01                   CMP     #1
1554  0791  D0 1C                   BNE     RNGERR          ;IF >1, BAD
```

```
1555 0793 F0 04          BEQ     RANGE2          ;IF 1, GO CHECK LOW BYTE
1556 0795 A5 01       C  RANGE1: LDA     COLCRS+1     ;FOR OTHERS, NON-ZERO HIGH BYTE IS BAD
1557 0797 D0 16          BNE     RNGERR
1558 0799 BD 0B 0C    F  RANGE2: LDA     COLUMN,X     ;CHECK LOW BYTE
1559 079C C5 00       E          CMP     COLCRS
1560 079E 90 0F          BCC     RNGERR
1561 07A0 F0 0D          BEQ     RNGERR
1562 07A2 A9 01          RNGOK   LDA     #SUCCES      ;SET STATUS OK
1563 07A4 85 00       E          STA     DSTAT
1564 07A6 A9 80                  LDA     #BRKABT      ;PREPARE BREAK ABORT STATUS
1565 07A8 A6 00       E          LDX     BRKKEY       ;CHECK BREAK KEY FLAG
1566 07AA 85 00       E          STA     BRKKEY       ;'CLEAR' BREAK
1567 07AC F0 06                  BEQ     RNGER2       ;IF BREAK, QUIT IMMEDIATELY AND RETURN TO CI
1568 07AE 60                     RTS
1569 07AF 20 F4 04    P  RNGERR  JSR     HOME         ;ON RANGE ERROR, BRING CURSOR BACK
1570 07B2 A9 8D                  LDA     #CRSROR      ;SHOW CURSOR OVERRANGE ERROR
1571 07B4 85 00       E  RNGER2: STA     DSTAT
1572 07B6 68                     RNGER1: PLA                  ;RESTORE STACK (THIS ROUTINE IS ALWAYS 1 LEV
1573 07B7 68                             PLA                  ;AWAY FROM RETURN TO CIO)
1574 07B8 A5 00       E                  LDA     SWPFLG       ;IF SWAPPED, SWAP BACK
1575 07BA 10 03                          BPL     RETUR3
1576 07BC 4C 00 0A    F                  JMP     SWAP
1577 07BF 4C A2 02    F  RETUR3: JMP     RETUR1       ;RETURN TO CIO
1578                             ;
1579                             ;
1580                             ;
1581                             ;  OFFCRS: RESTORE OLD DATA UNDER CURSOR SO IT CAN BE MOVED
1582                             ;
1583 07C2 A0 00          OFFCRS: LDY     #0
1584 07C4 A5 01       E          LDA     OLDADR+1
1585 07C6 F0 04                  BEQ     OFF090       ;DON'T STORE TO PAGE 0
1586 07C8 A5 00       E          LDA     OLDCHR
1587 07CA 91 00       E          STA     (OLDADR),Y
1588 07CC 60             OFF090: RTS
1589                             ;
1590                             ;
1591                             ;
1592                             ;  BITMAP ROUTINES:
1593                             ;
1594                             ;  BITCON  PUT MASK IN BITMSK AND INDEX IN X
1595                             ;  BITPUT  PUT CARRY INTO BITMAP
1596                             ;  BITROL  ROL CARRY INTO BOTTOM OF BITMAP (SCROLL)
1597                             ;  BITSET  SET PROPER BIT
1598                             ;  BITCLR  CLEAR PROPER BIT
1599                             ;  BITGET  RETURN CARRY SET IF BIT IS THERE
1600                             ;  LOGGET  DO BITGET FOR LOGMAP INSTEAD OF TABMAP
1601                             ;
1602 07CD 48             BITCON: PHA
1603 07CE 29 07                  AND     #7
1604 07D0 AA                     TAX                  ;GET MASK
1605 07D1 BD 42 0C    P          LDA     MASKTB,X
1606 07D4 85 00       E          STA     BITMSK
1607 07D6 68                     PLA                  ;PROCESS INDEX
1608 07D7 4A                     LSR     A
```

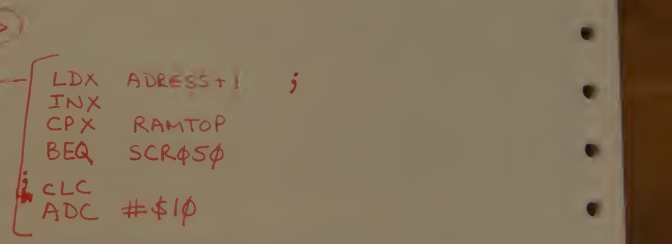
```
1609  07D8  4A              LSR   A
1610  07D9  4A              LSR   A
1611  07DA  AA              TAX
1612  07DB  60              RTS
1613
1614
1615  07DC  2E 02 00    E   BITROL  ROL   LOGMAP+2
1616  07DF  2E 01 00    E           ROL   LOGMAP+1
1617  07E2  2E 00 00    E           ROL   LOGMAP
1618  07E5  60                      RTS
1619
1620
1621  07E6  90 0C           BITPUT  BCC   BITCLR       ;AND RETURN
1622                        ; OTHERWISE FALL THROUGH TO BITSET AND RETURN
1623  07E8  20 CD 07    P   BITSET  JSR   BITCON
1624  07EB  BD 00 00    E           LDA   TABMAP,X
1625  07EE  05 00       E           ORA   BITMSK
1626  07F0  9D 00 00    E           STA   TABMAP,X
1627  07F3  60                      RTS
1628
1629  07F4  20 CD 07    P   BITCLR  JSR   BITCON
1630  07F7  A5 00       E           LDA   BITMSK
1631  07F9  49 FF                   EOR   #$FF
1632  07FB  3D 00 00    E           AND   TABMAP,X
1633  07FE  9D 00 00    E           STA   TABMAP,X
1634  0801  60                      RTS
1635
1636  0802  A5 00       E   LOGGET  LDA   ROWCRS
1637  0804  18              LOIGET  CLC
1638  0805  69 78           LO2GET  ADC   #120         ;TABMAP TO LOGMAP OFFSET
1639  0807  20 CD 07    P   BITGET  JSR   BITCON
1640  080A  18                      CLC
1641  080B  BD 00 00    E           LDA   TABMAP,X
1642  080E  25 00       E           AND   BITMSK
1643  0810  F0 01                   BEQ   BITGE1
1644  0812  38                      SEC
1645  0813  60              BITGE1  RTS
1646
1647
1648
1649
1650                        ; INATAC: INTERNAL(CHAR) TO ATASCII(ATACHR) CONVERSION
1651
1652  0814  A0 00 00    E   INATAC  LDA   CHAR
1653  0817  A4 00       E           LDY   DINDEX       ;IF GRAPHICS MODES
1654  0819  C0 0F                   CPY   #15
1655  081B  B0 17                   BCS   INATA1       ;MODE >= 15,
1656  081D  C0 0C                   CPY   #12
1657  081F  B0 04                   BCS   INATA0       ;MODE = 12-14,
1658  0821  C0 03                   CPY   #3           ;MODE >= 3
1659  0823  B0 0F                   BCS   INATA1       ;THEN DON'T CHANGE CHAR
1660  0825  2A              INATA0  ROL   A
1661  0826  2A                      ROL   A
1662  0827  2A                      ROL   A
```

```
1663  0828  0A                    REL     A
1664  0829  29 03                 AND     #3
1665  082B  AA                    TAX
1666  082C  AD 00 00        E     LDA     CHAR
1667  082F  29 9F                 AND     #$9F
1668  0831  1D 73 0C        P     ORA     INTATA,X
1669  0834  8D 00 00        E     STA     ATACHR
1670  0837  60              MOV090: RTS
1671
1672
1673                        ; MOVLIN
1674
1675  0838  A6 00           E     MOVLIN: LDX     RAMTOP          ;FRMADR := MEMTOP -80
1676  083A  CA                    DEX                             ;TOADR  := MEMTOP - 40
1677  083B  86 01           E     STX     FRMADR+1
1678  083D  8A 01           E     STX     TOADR+1
1679  083F  A9 80                 LDA     #$80            ;$XX00-80
1680  0841  85 00           E     STA     FRMADR
1681  0843  A9 D8                 LDA     #$D8            ;$XX00-40
1682  0845  85 00           E     STA     TOADR
1683
1684  0847  A6 00           E     LDX     ROWCRS
1685
1686  0849  E8              MOV020: INX                     ;DONE?
1687  084A  EC 00 00        E     CPX     BOTSCR
1688  084D  F0 EE                 BEQ     MOV090          ;YES
1689
1690  084F  A0 27                 LDY     #39
1691  0851  B1 00           E     MOV030: LDA     (FRMADR),Y      ;MOVE DATA
1692  0853  91 00           E     STA     (TOADR),Y
1693  0855  88                    DEY
1694  0856  10 F9                 BPL     MOV030
1695
1696  0858  38                    SEC
1697  0859  A5 00           E     LDA     FRMADR          ;TOADR := FRMADR
1698  085B  85 00           E     STA     TOADR
1699  085D  E9 28                 SBC     #40             ;FRMADR := FRMADR-40
1700  085F  85 00           E     STA     FRMADR
1701  0861  A5 01           E     LDA     FRMADR+1
1702  0863  85 01           E     STA     TOADR+1
1703  0865  E9 00                 SBC     #0
1704  0867  85 01           E     STA     FRMADR+1
1705  0869  4C 49 08        P     JMP     MOV020
1706
1707                        ; EXTEND  EXTEND BIT MAP FROM ROWCRS (EXTEND LOGICAL LINE
1708
1709  086C  08              EXTEND  PHP                     ;SAVE CARRY
1710  086D  A0 17                   LDY     #23
1711  086F  98              EXTEN1: TYA
1712  0870  20 04 08        P     JSR     LDIGET
1713  0873  08                    PHP
1714  0874  98                    TYA
1715  0875  18                    CLC
1716  0876  69 79                 ADC     #121
```

```
1717  0878  38              PLP
1718  0879  20 E6 07    P   JSR     BITPUT
1719  087C  88              DEY
1720  087D  30 04           BMI     EXTEN4
1721  087F  C4 D0        E  CPY     ROWCRS
1722  0881  90 E5           BCC     EXTEN1
1723  0883  A5 D0        E  EXTEN4: LDA     ROWCRS
1724  0885  18              CLC
1725  0886  69 78           ADC     #120
1726  0888  28              PLP
1727  0889  4C E6 07    P   JMP     BITPUT      ;STORE NEW LINE/0 BIT AND RETURN
1728                        ;
1729                        ; CLRLIN  CLEAR LINE CURSOR IS ON
1730                        ;
1731  088C  A5 00        E  CLRLIN: LDA     LMARGN
1732  088E  85 D0        E          STA     COLCRS
1733  0890  20 5A 06     P          JSR     CONVRT
1734  0893  38                     SEC
1735  0894  A5 00        E          LDA     RMARGN
1736  0896  E5 00        E          SBC     LMARGN
1737  0898  A8                     TAY
1738  0899  A9 00                  LDA     #0
1739  089B  91 00        E  CLRLI1: STA     (ADRESS),Y
1740  089D  88                     DEY
1741  089E  10 F8                  BPL     CLRLI1
1742  08A0  60                     RTS
1743                        ;
1744                        ; SCROLL  SCROLL SCREEN FROM BOTTOM TO TOP
1745                        ;
1746  08A1  20 DC 07    P  SCROLL: JSR     BITROL      ;ROLL IN CARRY
1747                        ;
1748  08A4  AD 00 00    E  SCR005: LDA     FINE        ;FINE SCROLLING?
1749  08A7  F0 2B                  BEQ     SCR019      ;NO
1750                        ;
1751  08A9  AD 00 00    E  SCR007: LDA     VSFLAG      ;JUST IN CASE!
1752  08AC  D0 F8                  BNE     SCR007      ;PRIOR SCROLL NOT DONE YET
1753                        ;
1754  08AE  A9 08                  LDA     #8          ;SET VERTICAL SCROLL FLAG
1755  08B0  8D 00 00    E          STA     VSFLAG
1756                        ;
1757  08B3  AD 00 00    E  SCR010: LDA     VSFLAG      ;WAIT FOR FINE SCROLL TO COMPLETE
1758  08B6  C9 01                  CMP     #1          ;WAITING FOR START OF LAST SCAN.
1759  08B8  D0 F9                  BNE     SCR010
1760                        ;
1761  08BA  AD 0B D4          SCR019: LDA     VCOUNT      ;WAIT FOR SAFE PLACE (START OF SCAN)
1762  08BD  C9 40                  CMP     #$40
1763  08BF  B0 F9                  BCS     SCR019
1764                        ;
1765  08C1  A2 00                  LDX     #$00
1766  08C3  AD 00 00    E          LDA     BOTSCR      ;SPLIT SCREEN?
1767  08C6  C9 04                  CMP     #4
1768  08C8  D0 02                  BNE     SCR018      ;NO
1769  08CA  A2 70                  LDX     #$70        ;YES
1770                        ;
```

```
ERR LINE  ADDR  B1 B2 B3 B4      DISPLAY HANDLER  -- 10-30-78 --  DISPLC                    PAGE  27

    1771  08CC  EC 08 D4         SCR018  CPX     VCOUNT      ;LET THE SCAN GET A BIT AHEAD
    1772  08CF  B0 F8                     BCS     SCR018
    1773                        ;
    1774  08D1  20 A4 0A      P  SCR019  JSR     PUTMSC      ;ADRESS := SAVMSC
    1775                        ;
    1776                        ; SCROLL SCREEN FOR DELETE
    1777                        ;
    1778  08D4  A5 00         E  SCRDLI  LDA     ADRESS      ;CALCULATE BYTES TO MOVE
    1779  08D6  A6 01         E          LDX     ADRESS+1
    1780                        ;
    1781  08D8  E8                 SCR300  INX
    1782  08D9  E4 00         E          CPX     RAMTOP
    1783  08DB  F0 06                    BEQ     SCR310
    1784                        ;
    1785  08DD  38                       SEC
    1786  08DE  E9 10                    SBC     #$10
    1787  08E0  4C D8 08      P          JMP     SCR300
    1788                        ;
    1789  08E3  69 27         SCR310  ADC     #39         ;(CLC) ADC #40)
    1790  08E5  F0 22                    BNE     SCR340      ;BYTE COUNT = 0
    1791                        ;
    1792  08E7  A8             SCR340: TAY                 ;Y= - # BYTES
    1793  08E8  85 00         E          STA     COUNTR
    1794  08EA  38                       SEC                 ;ADRESS =ADRESS-Y
    1795  08EB  A5 00         E          LDA     ADRESS
    1796  08ED  E5 00         E          SBC     COUNTR
    1797  08EF  85 00         E          STA     ADRESS
    1798  08F1  B0 02                    BCS     SCR020
    1799  08F3  C6 01         E          DEC     ADRESS+1
    1800                        ;
    1801  08F5  A5 00         E  SCR020: LDA     ADRESS      ;NOW MOVE THE DATA DOWN.
    1802  08F7  18                       CLC
    1803  08F8  69 28                    ADC     #40
    1804  08FA  85 00         E          STA     COUNTR      ;COUNTR := ADRESS + 40.
    1805  08FC  A5 01         E          LDA     ADRESS+1
    1806  08FE  69 00                    ADC     #0
    1807  0900  85 01         E          STA     COUNTR+1
    1808                        ;
    1809  0902  B1 00         E  SCR030: LDA     (COUNTR),Y  ;ADRESS(Y) := COUNTR(Y).
    1810  0904  91 00         E          STA     (ADRESS),Y
    1811  0906  C8                       INY
    1812  0907  D0 F9                    BNE     SCR030      ;256-16 TIMES
    1813                        ;
    1814  0909  A0 10           SCR040: LDY     #16         ;(-240)
    1815  090B  A5 00         E          LDA     ADRESS      ;ADRESS := ADRESS + 240.
    1816  090D  C9 D8                    CMP     #-40        ;ALL DONE?
    1817  090F  F0 0B                    BEQ     SCR050      ;YES
    1818                        ;
    1819  0911  18                       CLC
    1820  0912  69 F0                    ADC     #240
    1821  0914  85 00         E          STA     ADRESS
    1822  0916  90 00                    BCC     SCR020
    1823                        ;
    1824  0918  E6 01         E          INC     ADRESS+1
```



```
###  LINE  ADDR  01 02 03 04        DISPLAY HANDLER  -- 10-30-79 --  DISPLC                    PAGE  30

 1828  091A  D0 09                        BNE     SCR020          ;(JMP)
 1826
 1827  091C  18              SCR060:     CLC                     ;DATA ALL MOVED
 1828  091D  A5 00           E           LDA     COUNTR          ;COUNTR = COUNTR + 160+16+40
 1829  091F  69 D8                       ADC     #160+16+40
 1830  0921  85 00           E           STA     COUNTR          ;(DOESN'T GENERATE CARRY)
 1831  0923  A9 00                       LDA     #0              ;NOW CLEAR THE LAST LINE
 1832  0925  A0 27                       LDY     #39
 1833                        ;
 1834  0927  91 00           E   SCR060: STA     (COUNTR),Y
 1835  0929  88                         DEY
 1836  092A  10 FB                      BPL     SCR060
 1837                        ;
 1838                        ;           JMP     DOLCOL          ;AND RETURN
 1839                        ;
 1840                        ;
 1841                        ;
 1842                        ;  DOLCOL:  DO LOGICAL COLUMN FROM BITMAP AND COLCRS
 1843                        ;
 1844  092C  A9 00           DOLCOL:     LDA     #0              ;START WITH ZERO
 1845  092E  85 00           E           STA     LOGCOL
 1846  0930  A5 00           E           LDA     ROWCRS
 1847  0932  85 00           E           STA     HOLD1
 1848  0934  A5 00           E   DOLCO1: LDA     HOLD1           ;ADD IN ROW COMPONENT
 1849  0936  20 04 08        P           JSR     LO1GET
 1850  0939  B0 0C                      BCS     DOLCO2          ;FOUND BEGINNING OF LINE
 1851  093B  A5 00           E           LDA     LOGCOL          ;ADD 40 AND LOOK BACK ONE
 1852  093D  18                         CLC
 1853  093E  69 28                      ADC     #40
 1854  0940  85 00           E           STA     LOGCOL
 1855  0942  C6 00           E           DEC     HOLD1           ;UP ONE LINE
 1856  0944  4C 34 09        P           JMP     DOLCO1
 1857  0947  18              DOLCO2:     CLC                     ;ADD IN COLCRS
 1858  0948  A5 00           E           LDA     LOGCOL
 1859  094A  65 00           E           ADC     COLCRS
 1860  094C  85 00           E           STA     LOGCOL
 1861  094E  60                          RTS
 1862                        ;
 1863                        ;
 1864                        ;
 1865                        ;  DOBUFC:  COMPUTE BUFFER COUNT AS THE NUMBER OF BYTES FROM
 1866                        ;           BUFSTR TO END OF LOGICAL LINE WITH TRAILING SPACES REMOVED
 1867                        ;
 1868  094F  20 EA 09        P   DOBUFC: JSR     PHACRS
 1869  0952  A5 00           E           LDA     LOGCOL
 1870  0954  48                          PHA
 1871  0955  A5 00           E           LDA     BUFSTR          ;START
 1872  0957  85 00           E           STA     ROWCRS
 1873  0959  A5 01           E           LDA     BUFSTR+1
 1874  095B  85 00           E           STA     COLCRS
 1875  095D  A9 01                       LDA     #1
 1876  095F  85 00           E           STA     BUFCNT
 1877  0961  A2 17           DOBUF1:     LDX     #23             ;NORMAL
 1878  0963  A5 00           E           LDA     SWPFLG          ;IF SWAPPED, ROW 3 IS THE LAST LINE ON SCREE
```

LDX RAMTOP
DEX
STX COUNTR+1
LDX #-40
STA COUNTR

```
1879  0965 10 02        E       BPL     DOBI
1880  0967 A2 03        E       LDX     #3
1881  0969 E4 00        E DOBI  CPX     ROWCRS          ;TEST IF CRSR IS AT LAST SCREEN POSITION
1882  096B D0 08        E       BNE     DOBUIA
1883  096D A5 00        E       LDA     COLCRS
1884  096F C5 00        E       CMP     RMARGN
1885  0971 D0 05        E       BNE     DOBUIA
1886  0973 E6 00        E       INC     BUFCNT          ;YES, SO FAKE INCRSR TO AVOID SCROLLING
1887  0975 4C 88 09     P       JMP     DOBUF2
1888  0978 20 B4 06     P DOBUIA JSR    INCRSR
1889  097B E6 00        E       INC     BUFCNT
1890  097D A5 00        E       LDA     LOGCOL
1891  097F C5 00        E       CMP     LMARGN
1892  0981 00 DE        E       BNE     DOBUF1          ;NOT YET BOL
1893  0983 C6 00        E       DEC     ROWCRS          ;BACK UP ONE INCRSR
1894  0985 20 B4 04     P       JSR     CRSRLF
1895  0988 20 13 02     P DOBUF2 JSR    GETFLT          ;TEST CURRENT COLUMN FOR NON-ZERO DATA
1896  098B D0 17        E       BNE     DOBUF4          ;QUIT IF NON-ZERO
1897  098D C6 00        E       DEC     BUFCNT          ;DECREMENT COUNTER
1898  098F A5 00        E       LDA     LOGCOL          ;BEGINNING OF LOGICAL LINE YET?
1899  0991 C5 00        E       CMP     LMARGN
1900  0993 F0 0F        E       BEQ     DOBUF4          ;YES, SO QUIT
1901  0995 20 B4 04     P       JSR     CRSRLF          ;BACK UP CURSOR
1902  0998 A5 00        E       LDA     COLCRS          ;IF LOGCOL=RMARGN, GO UP 1 ROW
1903  099A C5 00        E       CMP     RMARGN
1904  099C D0 02        E       BNE     DOBUF3
1905  099E C6 00        E       DEC     ROWCRS
1906  09A0 A5 00        E DOBUF3 LDA    BUFCNT
1907  09A2 D0 E4        E       BNE     DOBUF2          ;LOOP UNLESS BUFCNT JUST WENT TO ZERO
1908  09A4 68           E DOBUF4 PLA
1909  09A5 85 00        E       STA     LOGCOL
1910  09A7 4C F5 09     P       JMP     PLACRS          ;AND RETURN
1911                              ;
1912                              ;
1913                              ;
1914                              ;
1915                              ; STRBEG  MOVE BUFSTR TO BEGINNING OF LOGICAL LINE
1916                              ;
1917  09AA 20 2C 09     P STRBEG JSR    DOLCOL          ;USE DOLCOL TO POINT HOLD1 AT BOL
1918  09AD A5 00        E       LDA     HOLD1
1919  09AF 85 00        E       STA     BUFSTR
1920  09B1 A5 00        E       LDA     LMARGN
1921  09B3 85 01        E       STA     BUFSTR+1
1922  09B5 60           E DELTI3 RTR
1923                              ;
1924                              ;
1925                              ;
1926                              ;
1927                              ;
1928                              ; DELTIM  TIME TO DELETE A LINE IF IT IS EMPTY AND AN EXTENSION
1929                              ;
1930  09B6 A5 00        E DELTIA LDA    LOGCOL          ;IF LOGCOL<>LMARGN
1931  09B8 C5 00        E       CMP     LMARGN          ;THEN DONT MOVE UP ONE
1932  09BA D0 02        E       BNE     DELTIB          ;LINE BEFORE TESTING DELTIM
```

```
1933  09BB  E6 BB      E          INC     ROWCRS
1934  09BE  20 2C 05   F  DELTIB  JSR     GOLOGL
1935  09C1  A5 00      E  DELTIM  LDA     LOGICAL         ;TEST FOR EXTENSION
1936  09C3  C5 C0      E          CMP     LMARGN
1937  09C5  F0 BE               BEQ     DELTI3          ;NO
1938  09C7  20 96 06   F          JSR     CONVRT
1939  09CA  A5 00      E          LDA     RMARGN          ;SET UP COUNT
1940  09CC  38                   SEC
1941  09CD  E5 00      E          SBC     LMARGN
1942  09CF  A8                   TAY
1943  09D0  B1 00      E  DELTI1  LDA     (ADRESS),Y
1944  09D2  D0 E1               BNE     DELTI3          ;FOUND A NON-0 SO QUIT AND RETURN
1945  09D4  88                   DEY
1946  09D5  10 F9               BPL     DELTI1
1947  09D7  4C D8 D5   F  DELTI2  JMP     DELLIB          ;DELETE A LINE AND RETURN
1948                             ;
1949                             ;
1950                             ;
1951                             ; TSTCTL  SEARCH CNTRLS TABLE TO SEE IF ATACHR IS A CNTL CHAR
1952                             ;
1953  09DA  A2 2D               TSTCTL  LDX     #45             ;PREPARE TO SEARCH TABLE
1954  09DC  BD 53 06   P        TSTCT1  LDA     CNTRLS,X
1955  09DF  CD 00 00   E                CMP     ATACHR
1956  09E2  F0 05               BEQ     TSTCT2
1957  09E4  CA                   DEX
1958  09E5  CA                   DEX
1959  09E6  CA                   DEX
1960  09E7  10 F3               BPL     TSTCT1
1961  09E9  60                 TSTCT2  RTS
1962                             ;
1963                             ;
1964                             ;
1965                             ; PUSH ROWCRS,COLCRS AND COLCRS+1
1966                             ;
1967  09EA  A2 02              PHACRS  LDX     #2
1968  09EC  B5 00      E       PHACR1  LDA     ROWCRS,X
1969  09EE  9D 00 00   E               STA     TMPROW,X
1970  09F1  CA                   DEX
1971  09F2  10 F8               BPL     PHACR1
1972  09F4  60                   RTS
1973                             ;
1974                             ;
1975                             ; PULL COLCRS+1,COLCRS AND ROWCRS
1976                             ;
1977  09F5  A2 02              PLACRS  LDX     #2
1978  09F7  BD 00 00   E       PLACR1  LDA     TMPROW,X
1979  09FA  95 00      E               STA     ROWCRS,X
1980  09FC  CA                   DEX
1981  09FD  10 F8               BPL     PLACR1
1982  09FF  60                   RTS
1983                             ;
1984                             ;
1985                             ;
1986                             ; SWAP  IF MIXED MODE, SWAP TEXT CURSORS WITH REGULAR CURSORS
```

```
1987                    ;
1988 0A00 AD 00 00   E  SWAP    LDA     BOTSCR
1989 0A03 C9 18          CMP     #24
1990 0A05 F0 17          BEQ     SWAP3
1991 0A07 A2 0B          LDX     #11
1992 0A09 B5 00      E   SWAP1   LDA     ROWCRS,X
1993 0A0B 48             PHA
1994 0A0C BD 00 00   E           LDA     TXTROW,X
1995 0A0F 95 00      E           STA     ROWCRS,X
1996 0A11 68             PLA
1997 0A12 9D 00 00   E           STA     TXTROW,X
1998 0A15 CA             DEX
1999 0A16 10 F1          BPL     SWAP1
2000 0A18 A5 00      E           LDA     SWPFLG
2001 0A1A 49 FF          EOR     #$FF
2002 0A1C 85 00      E           STA     SWPFLG
2003 0A1E 4C A2 02   P   SWAP3   JMP     RETUR1
2004                    ;
2005                    ;
2006                    ; CLICK  MAKE CLICK THROUGH KEYBOARD SPEAKER
2007                    ;
2008 0A21 A2 7F          CLICK   LDX     #$7F
2009 0A23 48             PHA
2010 0A24 8E 1F D0       CLICK1: STX     CONSOL
2011 0A27 AD 0B D4               LDA     VCOUNT
2012 0A2A CD 0B D4       CLICK2: CMP     VCOUNT
2013 0A2D F0 FB                  BEQ     CLICK2      ;WAIT FOR NEXT SCAN
2014 0A2F CA                     DEX
2015 0A30 CA                     DEX
2016 0A31 10 F1                  BPL     CLICK1
2017 0A33 68                     PLA
2018 0A34 60                     RTS
2019                    ;
2020                    ;
2021                    ; COLCR  PUTS EITHER 0 OR LMARGN INTO COLCRS BASED ON MODE AND SWPFLG
2022                    ;
2023 0A35 A9 00          COLCR   LDA     #0
2024 0A37 A6 00      E           LDX     SWPFLG
2025 0A39 D0 04                  BNE     COLCR1
2026 0A3B A6 00      E           LDX     DINDEX
2027 0A3D D0 02                  BNE     COLCR2
2028 0A3F A5 00      E   COLCR1: LDA     LMARGN
2029 0A41 85 00      E   COLCR2: STA     COLCRS
2030 0A43 60                     RTS
2031                    ;
2032                    ;
2033                    ; PUTMSC  PUT SAVMSC INTO ADRESS
2034                    ;
2035 0A44 A5 00      E   PUTMSC: LDA     SAVMSC      ;SET UP ADDRESS
2036 0A46 85 00      E           STA     ADRESS
2037 0A48 A5 01      E           LDA     SAVMSC+1
2038 0A4A 85 01      E           STA     ADRESS+1
2039 0A4C 60                     RTS
2040                    ;
```

```
                        PAGE

                ; DRAW -- DRAW A LINE FROM OLDROW,OLDCOL TO NEWROW,NEWCOL
                ; (THE AL KILLER METHOD FROM BASKETBALL)
0A46 A2 00      DRAW    LDX     #0
0A48 AD 00 00   E       LDA     ICCOMZ          ;TEST COMMAND  $11=DRAW $12=FILL
0A4B C9 11              CMP     #$11
0A4D F0 08              BEQ     DRAWA
0A4F C9 12              CMP     #$12            ;TEST FILL
0A51 F0 03              BEQ     DRAWB           ;YES
0A53 A0 B4              LDY     #INVALID        ;NO, SO RETURN INVALID COMMAND
0A55 60                 RTS
0A56 E8         DRAWB   INX
0A57 8E 00 00   E       STX     FILFLG
0A5A A5 00      E       LDA     ROWCRS          ;PUT CURSOR INTO NEWROW,NEWCOL
0A5C 8D 00 00   E       STA     NEWROW
0A5F A5 00      E       LDA     COLCRS
0A61 8D 00 00   E       STA     NEWCOL
0A64 A5 01      E       LDA     COLCRS+1
0A66 8D 01 00   E       STA     NEWCOL+1
0A69 A9 01              LDA     #1
0A6B 8D 00 00   E       STA     ROWINC          ;SET UP INITIAL DIRECTIONS
0A6E 8D 00 00   E       STA     COLINC
0A71 38                 SEC
0A72 AD 00 00   E       LDA     NEWROW          ;DETERMINE DELTA ROW
0A75 E5 00      E       SBC     OLDROW
0A77 85 00      E       STA     DELTAR
0A79 B0 0E              BCS     DRAW1           ;DO DIRECTION AND ABSOLUTE VALUE
0A7B A9 FF              LDA     #$FF            ;BORROW WAS ATTEMPTED
0A7D 8D 00 00   E       STA     ROWINC          ;SET DIRECTION=DOWN
0A80 A5 00      E       LDA     DELTAR
0A82 49 FF              EOR     #$FF            ;DELTAR = |DELTAR|
0A84 18                 CLC
0A85 69 01              ADC     #1
0A87 85 00      E       STA     DELTAR
0A89 38         DRAW1   SEC
0A8A AD 00 00   E       LDA     NEWCOL          ;NOW DELTA COLUMN
0A8D E5 00      E       SBC     OLDCOL
0A8F 85 00      E       STA     DELTAC
0A91 AD 01 00   E       LDA     NEWCOL+1        ;TWO-BYTE QUANTITY
0A94 E5 01      E       SBC     OLDCOL+1
0A96 85 01      E       STA     DELTAC+1
0A98 B0 17              BCS     DRAW2           ;DIRECTION AND ABSOLUTE VALUE
0A9A A9 FF              LDA     #$FF            ;BORROW WAS ATTEMPTED
0A9C 8D 00 00   E       STA     COLINC          ;SET DIRECTION = LEFT
0A9F A5 00      E       LDA     DELTAC
0AA1 49 FF              EOR     #$FF            ;DELTAC = |DELTAC|
0AA3 85 00      E       STA     DELTAC
0AA5 A5 01      E       LDA     DELTAC+1
0AA7 49 FF              EOR     #$FF
0AA9 85 01      E       STA     DELTAC+1
0AAB E6 00      E       INC     DELTAC          ;ADD ONE FOR TWOS COMPLEMENT
0AAD D0 02              BNE     DRAW2
```

```
                0A85 E6 D1      E       INC     DELTAC+1
                0A87 A2 02              DRAW2   LDX     #2              ;ZERO RAM FOR DRAW LOOP
                0A89 A0 00              LDY     #0
                0A8B 94 01      E       STY     COLAC+1
                0A8D 98                 DRAW3A  TYA
                0A8E 95 00      E       STA     ROWAC,X
                0A90 B5 00      E       LDA     OLDROW,X
                0AC2 95 00      E       STA     ROWCRS,X
                0AC4 CA                 DEX
                0AC5 10 F6              BPL     DRAW3A
                0AC7 A5 00      E       LDA     DELTAC          ;FIND LARGER ONE (ROW OR COL)
                                        STA     COUNTR          ;PREPARE COUNTR AND ENDPT)
                                        STA     ENDPT
                0AC9 E8                 INX                     ;MAKE X 0
                0ACA A8                 TAY
                0ACB A5 01      E       LDA     DELTAC+1
                0ACD 85 01      E       STA     COUNTR+1
                0ACF 85 01      E       STA     ENDPT+1
                0AD1 D0 08              BNE     DRAW3           ;AUTOMATICALLY LARGER IF MSD>0
                0AD3 A5 00      E       LDA     DELTAC
                0AD5 C5 00      E       CMP     DELTAR          ;LOW COL ?LOW ROW?
                0AD7 B0 09              BCS     DRAW3           ;YES
                0AD9 A5 00      E       LDA     DELTAR
                0ADB A2 02              LDX     #2
                0ADD A8                 TAY
                0ADE 98                 DRAW3   TYA             ;PUT IN INITIAL CONDITIONS
                0ADF 85 00      E       STA     COUNTR
                0AE1 85 00      E       STA     ENDPT
                0AE3 48                 PHA                     ;SAVE AC
                0AE4 A5 01      E       LDA     ENDPT+1         ;PUT LSB OF HIGH BYTE
                0AE6 4A                 LSR     A               ;INTO CARRY
                0AE7 68                 PLA                     ;RESTORE AC
                0AE8 6A                 ROR     A               ;ROR THE 9 BIT ACCUMULATOR
                0AE9 95 00      E       STA     ROWAC,X
                0AEB A5 00      E       DRAW4A  LDA     COUNTR  ;TEST ZERO
                0AED 05 01      E               ORA     COUNTR+1
                0AEF D0 03              BNE     DRAW11          ;IF COUNTER IS ZERO, LEAVE DRAW
                0AF1 4C 9F 0B   F       JMP     DRAW10
                0AF4 18                 DRAW11  CLC             ;ADD ROW TO ROWAC (PLOT LOOP)
                0AF5 A5 00      E       LDA     ROWAC
                0AF7 65 00      E       ADC     DELTAR
                0AF9 85 00      E       STA     ROWAC
                0AFB 90 02              BCC     DRAW5
                0AFD E6 01      E       INC     ROWAC+1
                0AFF A5 01      E       DRAW5   LDA     ROWAC+1 ;COMPARE ROW TO ENDPOINT
                0B01 C5 01      E       CMP     ENDPT+1         ;IF HIGH BYTE OF ROW IS .LT. HIGH
                0B03 90 15              BCC     DRAW6           ;BYTE OF ENDPT, BLT TO COLUMN
                0B05 D0 06              BNE     DRAW5A
                0B07 A5 00      E       LDA     ROWAC
                0B09 C5 00      E       CMP     ENDPT           ;LOW BYTE
                0B0B 90 0D              BCC     DRAW6           ;ALSO BLT
                0B0D 18                 DRAW5A  CLC             ;OK SO MOVE POINT
                0B0E A5 00      E       LDA     ROWCRS
                0B10 6D 00 00   E       ADC     ROWINC
```

```
2144  0BC3  95 00      E          STA   ROWCRS
2150  0BC5  A2 00      E          LDX   #0          ;AND SUBTRACT ENDPT FROM ROWAC
2151  0BC7  20 59 07   P          JSR   SUBEND
2152  0BCA  18                    CLC               ;DO SAME FOR COLUMN (DOUBLE BYTE ADD)
2153  0BCB  A5 00      E          LDA   COLAC       ;ADD
2154  0BCD  65 00      E          ADC   DELTAC
2155  0BCF  85 00      E          STA   COLAC
2156  0BD1  A5 01      E          LDA   COLAC+1
2157  0BD3  65 01      E          ADC   DELTAC+1
2158  0BD5  85 01      E          STA   COLAC+1
2159  0BD7  C5 01      E          CMP   ENDPT+1     ;COMPARE HIGH BYTE
2160  0BD9  90 28                BCC   DRAW8
2161  0BDB  D0 06                BNE   DRAW6A
2162  0BDD  A5 00      E          LDA   COLAC       ;COMPARE LOW BYTE
2163  0BDF  C5 00      E          CMP   ENDPT
2164  0BE1  90 20                BCC   DRAW8
2165  0BE3  2C 00 00   E  DRAW6A  BIT   COLINC      ;+ OR - ?
2166  0BE6  10 10                BPL   DRAW6B
2167  0BE8  C6 0D      E          DEC   COLCRS      ;DO DOUBLE BYTE DECREMENT
2168  0BEA  A5 00      E          LDA   COLCRS
2169  0BEC  C9 FF                CMP   #$FF
2170  0BEE  D0 0E                BNE   DRAW7
2171  0BF0  A5 01      E          LDA   COLCRS+1
2172  0BF2  F0 0A                BEQ   DRAW7       ;DON'T DEC IF ZERO
2173  0BF4  C6 01      E          DEC   COLCRS+1
2174  0BF6  10 06                BPL   DRAW7       ;(UNCONDITIONAL)
2175  0BF8  E6 00      E  DRAW6B  INC   COLCRS      ;DO DOUBLE BYTE INCREMENT
2176  0BFA  D0 02                BNE   DRAW7
2177  0BFC  E6 01      E          INC   COLCRS+1
2178  0BFE  A2 02              DRAW7  LDX   #2          ;AND SUBTRACT ENDPT FROM COLAC
2179  0C00  20 58 07   P          JSR   SUBEND
2180  0C03  20 74 07   P  DRAW8   JSR   RANGE
2181  0C06  20 4E 02   P          JSR   OUTPLT      ;PLOT POINT
2182  0C09  AD 00 00   E          LDA   FILFLG      ;TEST RIGHT FILL
2183  0C0C  F0 2F                BEQ   DRAW9
2184  0C0E  20 EA 09   P          JSR   PHACRS
2185  0C11  AD 00 00   E          LDA   ATACHR
2186  0C14  8D 00 00   E          STA   HOLD4
2187  0C17  A5 00      E  DRAW8A  LDA   ROWCRS      ;SAVE ROW IN CASE OF CR
2188  0C19  48                    PHA
2189  0C1A  20 BC 06   P          JSR   INCRSA      ;POSITION CURSOR ONE PAST DOT
2190  0C1D  68                    PLA               ;RESTORE ROWCRS
2191  0C1E  85 00      E          STA   ROWCRS
2192  0C20  20 74 07   P  DRAW8C  JSR   RANGE
2193  0C23  20 13 02   P          JSR   GETPLT      ;GET DATA
2194  0C26  D0 0C                BNE   DRAW8B      ;STOP IF NON-ZERO DATA IS ENCOUNTERED
2195  0C28  AD 00 00   E          LDA   FILDAT      ;FILL DATA
2196  0C2B  8D 00 00   E          STA   ATACHR
2197  0C2E  20 4E 02   P          JSR   OUTPLT      ;DRAW IT
2198  0C31  4C 67 0B   P          JMP   DRAW8A      ;LOOP
2199  0C34  AD 00 00   E  DRAW8B  LDA   HOLD4
2200  0C37  8D 00 00   E          STA   ATACHR
2201  0C3A  20 F5 09   P          JSR   PLACRS
2202  0C40  38                DRAW9  SEC               ;DO DOUBLE BYTE SUBTRACT
```

```
2203  088E  A5 00      E        LDA   COUNTR
2204  0890  E9 01               SBC   #1
2205  0892  85 00      E        STA   COUNTR
2206  0894  A5 01      E        LDA   COUNTR+1
2207  0896  E9 00               SBC   #0
2208  0898  85 01      E        STA   COUNTR+1
2209  089A  30 03               BMI   DRAW10
2210  089C  4C EB 0A   P        JMP   DRAW4A
2211  089F  4C A2 02   P  DRAW10: JMP   RETUR1
```

```
                                        ;
                                        ;  TABLES:
                                        ;
                                        ;  MEMORY ALLOCATION: (INCLUDING SPLITSCREEN WINDOW
                                        ;
0A00 08A2  18 10 0A 0A      ALOCAT  BYTE   24,16,10,10,10,20,50,100,196,196,196,196
0A04 08A6  10 1E 34 64
0A08 08AA  C4 C4 C4 C4
0A0C 08AE  17 1C 10 64              BYTE   93,28,16,100,196
0A11 08B2  C4
                                        ;
                                        ;  NUMBER OF DISPLAY LIST ENTRIES FOR MAIN SCREEN
                                        ;
0A14 08B3  17 17 0B 17      NUMDLE  BYTE   23,23,11,23,47,47,95,95,97,97,97,97
0A18 08B7  2F 2F 5F 5F
0A1C 08BB  61 61 61 61
0A20 08BF  12 17 09 0F              BYTE   18,23,11,191,97
0A24 08C3  61
0A25 08C4  13 13 09 13      MXDMDE  BYTE   19,19,9,19,39,39,79,79,65,65,65,65 ;(EXT OF NUMDLE)
0A29 08C8  27 27 4F 4F
0A2D 08CC  41 41 41 41
0A31 08D0  0F 13 09 9F              BYTE   15,19,9,159,65
0A35 08D4  41
                                        ;
                                        ;  ANTIC CODE FROM INTERNAL MODE CONVERSION TABLE
                                        ;
                                        ;    INTERNAL          ANTIC CODE              DESCRIPTION
                                        ;       0                 2                 40X2X8  CHARACTERS
                                        ;       1                 6                 20X5X8    ""
                                        ;       2                 7                 20X5X16   ""
                                        ;       3                 8                 40X4X8  GRAPHICS
                                        ;       4                 9                 80X2X4    ""
                                        ;       5                 A                 80X4X4    ""
                                        ;       6                 B                 160X2X2   ""
                                        ;       7                 D                 160X4X2   ""
                                        ;       8                 F                 320X2X1   ""
                                        ;       9                 SAME AS 8 BUT GTIA 'LUM' MODE
                                        ;       10                SAME AS 8 BUT GTIA 'COL/LUM REGISTER' MODE
                                        ;       11                SAME AS 8 BUT GTIA 'COLOR' MODE
                                        ;       12                3                 40X2X10 CHARACTERS
                                        ;       13                4                 40X5X8    ""
                                        ;       14                5                 40X5X16   ""
                                        ;       15                C                 160X2X1 GRAPHICS
                                        ;       16                E                 160X4X1   ""
                                        ;
0A39 08D5  22 06 07 08      ANCONV  BYTE   $22,6,7,8,9,$A,$B,$D,$F,$F,$F,$F ;ZEROS FOR RANGE TEST 1
0A3D 08D9  09 0A 0B 0D
0A41 08DD  0F 0F 0F 0F
0A45 08E1  03 04 05 0C              .BYTE  3,4,5,$C,$E
```

```
2266  0853  0C
2267
2268
2269                  ; PAGE TABLE TELLS WHICH DISPLAY LISTS ARE IN DANGER OF
2270                  ; CROSSING A 256 BYTE PAGE BOUNDARY
2271                  ;
2272  0854  00 00 00 00   PAGETB  BYTE  0,0,0,0,0,0,0,1,1,1,1,1
2273  0858  00 00 00 01
2274  085C  01 01 01 01
2275  0860  00 00 00 01         BYTE  0,0,0,1,1
2276  0864  01
2277                  ;
2278                  ; THIS IS THE NUMBER OF LEFT SHIFTS NEEDED TO MULTIPLY
2279                  ; COLCRS BY # BYTES/ROW   (ROWCRS*5)/(2**DMLINE)
2280                  ;
2281  0865  03 02 02 01   DMLINE  BYTE  3,2,2,1,1,2,2,3,3,3,3,3
2282  0869  01 02 02 03
2283  086D  03 03 03 03
2284  0871  03 03 03 02         BYTE  3,3,3,2,3
2285  0875  03
2286                  ;
2287                  ;
2288                  ; COLUMN  NUMBER OF COLUMNS
2289                  ;
2290  0876  28 14 14 28   COLUMN  BYTE  40,20,20,40,80,80,160,160, LOW 320,80,80,80 ;MODE 8 TO F
2291  087A  50 50 A0 A0
2292  087E  40 50 50 50
2293  0882  28 28 28 A0         BYTE  40,40,40,160,160
2294  0886  A0
2295                  ;
2296                  ;
2297                  ;
2298                  ; NOROWS  NUMBER OF ROWS
2299                  ;
2300  0887  18 18 0C 18   NOROWS  BYTE  24,24,12,24,48,48,96,96,192,192,192,192
2301  088B  30 30 60 60
2302  088F  C0 C0 C0 C0
2303  0893  13 18 0C C0         BYTE  19,24,12,192,192
2304  0897  C0
2305                  ;
2306                  ;
2307                  ;
2308                  ;
2309                  ; DIV2TX  HOW MANY RIGHT SHIFTS FOR HORSR FOR PARTIAL BYTE MODES
2310                  ;
2311  0898  00 00 00 02   DIV2TB  BYTE  0,0,0,2,3,2,3,2,3,1,1,1
2312  089C  02 02 03 02
2313  08A0  03 01 01 01
2314  08A4  00 00 00 02         BYTE  0,0,0,2,2
2315  08A8  02
2316                  ;
2317                  ;
2318                  ; DMASKT  DISPLAY MASK TABLE
2319
```

```
2320
2321  0C38  FF F0 0F          DMASKT  BYTE   $FF,$F0,$0F
2322  0C3F  C0 30 0C 03               BYTE   $C0,$30,$0C,$03
2323
2324                          ; MASKTB  BIT MASK   (ALSO PART OF DMASKTB! DO NOT SEPARATE!
2325
2326  0C42  80 40 20 10       MASKTB  BYTE   $80,$40,$20,$10,$08,$04,$02,$01
2327  0C46  08 04 02 01
2328
2329
2330
2331  0C4A  00 01 03 07       HMASK   BYTE   0,1,3,7
2332
2333
2334
2335
2336  0C4E  28 CA 94 46       COLRTB: BYTE   $28,$CA,$94,$46,$00
2337  0C52  00
2338
2339
2340
2341
2342                          ; CNTRLS: CONTROL CODES AND THEIR DISPLACEMENTS INTO THE
2343                          ;         CONTROL CHARACTER PROCESSORS
2344
2345  0C53  1B                CNTRLS  BYTE   $1B
2346  0C54  94 04         P           WORD   ESCAPE
2347  0C56  1C                        BYTE   $1C
2348  0C57  9A 04         P           WORD   CRSRUP
2349  0C59  1D                        BYTE   $1D
2350  0C5A  A7 04         P           WORD   CRSRDN
2351  0C5C  1E                        BYTE   $1E
2352  0C5D  B4 04         P           WORD   CRSRLF
2353  0C5F  1F                        BYTE   $1F
2354  0C60  C5 04         P           WORD   CRSRRT
2355  0C62  7D                        BYTE   $7D
2356  0C63  D4 04         P           WORD   CLRSCR
2357  0C65  7E                        BYTE   $7E
2358  0C66  04 05         P           WORD   BS
2359  0C68  7F                        BYTE   $7F
2360  0C69  2E 05         P           WORD   TAB
2361  0C6B  9B                        BYTE   $9B
2362  0C6C  0B 07         P           WORD   DOCRWS
2363  0C6E  9C                        BYTE   $9C
2364  0C6F  D4 05         P           WORD   DELLIN
2365  0C71  9D                        BYTE   $9D
2366  0C72  C0 05         P           WORD   INSLIN
2367  0C74  9E                        BYTE   $9E
2368  0C75  4E 05         P           WORD   CLRTAB
2369  0C77  9F                        BYTE   $9F
2370  0C78  49 05         P           WORD   SETTAB
2371  0C7A  FD                        BYTE   $FD
2372  0C7B  0A 06         P           WORD   BELL
2373  0C7D  FE                        BYTE   $FE
```

```
 2374  0C7E  89 05        P          WORD   DELCHR
 2375  0C80  FF                     BYTE   $FF
 2376  0C81  93 05        P          WORD   INSCHR
 2377
 2378                               ;
                                    ; SUPER FUNCTIONS (SUPERF = 1)
 2379                               ;
 2380  0C83  1C                     SUPERS BYTE   $1C
 2381  0C84  13 06        P          WORD   CHOME
 2382  0C86  1D                     BYTE   $1D
 2383  0C87  1E D6        P          WORD   CHBOT
 2384  0C89  1E                     BYTE   $1E
 2385  0C8A  1C 06        P          WORD   CHLEFT
 2386  0C8C  1F                     BYTE   $1F
 2387  0C8D  1F 06        P          WORD   CHRIGT
 2388                               ;
 2389                               ;
 2390                               ;
 2391                               ;
 2392                               ;
 2393                               ; ATAINT  ATASCI TO INTERNAL TABLE
 2394                               ;
 2395  0C8F  40 00 20 60            ATAINT BYTE   $40,$00,$20,$60
 2396                               ;
 2397                               ;
 2398                               ; INTATA  INTERNAL TO ATASCI TABLE
 2399                               ;
 2400  0C93  20 40 00 60            INTATA BYTE   $20,$40,$00,$60
 2401                               ;
 2402                               ;
 2403                               ; ATASCI  ATASCII CONVERSION TABLE
 2404                               ;
 2405  0C97  6C 6A 3B 8A            ATASCI BYTE   $6C,$6A,$3B,$8A,$8B,$6B,$28,$2A ;LOWER CASE
 2406  0C9B  8B 6B 28 2A
 2407  0C9F  6F 80 70 75                   BYTE   $6F,$80,$70,$75,$9B,$69,$2D,$3D
 2408  0CA3  9B 69 2D 3D
 2409  0CA7
 2410  0CA7  76 80 63 8C                   BYTE   $76,$80,$63,$8C,$8D,$62,$78,$7A
 2411  0CAB  8D 62 78 7A
 2412  0CAF  34 80 33 36                   BYTE   $34,$80,$33,$36,$1B,$35,$32,$31
 2413  0CB3  1B 35 32 31
 2414  0CB7
 2415  0CB7  2C 20 2E 6E                   BYTE   $2C,$20,$2E,$6E,$80,$6D,$2F,$81
 2416  0CBB  80 6D 2F 81
 2417  0CBF  72 80 65 79                   BYTE   $72,$80,$65,$79,$7F,$74,$77,$71
 2418  0CC3  7F 74 77 71
 2419  0CC7
 2420  0CC7  39 80 30 37                   BYTE   $39,$80,$30,$37,$7E,$38,$3C,$3E
 2421  0CCB  7E 38 3C 3E
 2422  0CCF  66 68 64 80                   BYTE   $66,$68,$64,$80,$82,$67,$73,$61
 2423  0CD3  82 67 73 61
 2424  0CD7
 2425  0CD7
 2426  0CD7  4C 4A 3A BA                   BYTE   $4C,$4A,$3A,$BA,$B8,$4B,$5C,$5E ;UPPER CASE
 2427  0CDB  B8 4B 5C 5E
```

```
2425  0CDF  4F 80 50 55   BYTE  $4F,$80,$50,$55,$98,$49,$5F,$7C
2426  0CE3  98 49 5F 7C
2427  0CE7
2428  0CE7  56 80 43 8C   BYTE  $56,$80,$43,$8C,$8D,$42,$58,$5A
2429  0CEB  8D 42 58 5A
2430  0CEF  24 80 23 26   BYTE  $24,$80,$23,$26,$18,$25,$22,$21
2431  0CF3  18 25 22 21
2432  0CF7
2433  0CF7  98 20 50 4E   BYTE  $98,$20,$50,$4E,$80,$4D,$3F,$81
2434  0CFB  80 4D 3F 81
2435  0CFF  52 80 45 59   BYTE  $52,$80,$45,$59,$9F,$54,$57,$51
2436  0D03  9F 54 57 51
2437  0D07
2438  0D07  28 80 29 27   BYTE  $28,$80,$29,$27,$9C,$40,$7D,$9D
2439  0D0B  9C 40 7D 9D
2440  0D0F  46 48 44 80   BYTE  $46,$48,$44,$80,$83,$47,$53,$41
2441  0D13  83 47 53 41
2442  0D17
2443  0D17
2444  0D17  0C 0A 7B 86   BYTE  $0C,$0A,$7B,$86,$80,$0B,$1E,$1F ;CONTROL
2445  0D1B  80 0B 1E 1F
2446  0D1F  0F 80 10 15   BYTE  $0F,$80,$10,$15,$9B,$09,$1C,$1D
2447  0D23  9B 09 1C 1D
2448  0D27
2449  0D27  16 80 03 89   BYTE  $16,$80,$03,$89,$92,$02,$18,$1A
2450  0D2B  92 02 18 1A
2451  0D2F  80 80 85 80   BYTE  $80,$80,$85,$80,$1B,$80,$FD,$80
2452  0D33  1B 80 FD 80
2453  0D37
2454  0D37  0D 20 60 0E   BYTE  $0D,$20,$60,$0E,$80,$0D,$80,$81
2455  0D3B  80 0D 80 81
2456  0D3F  12 80 05 19   BYTE  $12,$80,$05,$19,$9E,$14,$17,$11
2457  0D43  9E 14 17 11
2458  0D47
2459  0D47  80 80 80 80   BYTE  $80,$80,$80,$80,$FE,$80,$7D,$FF
2460  0D4B  FE 80 7D FF
2461  0D4F  06 08 04 80   BYTE  $06,$08,$04,$80,$84,$07,$13,$01
2462  0D53  84 07 13 01
2463
2464  0D57  1C 1D 1E 1F   FUNCKY BYTE  $1C,$1D,$1E,$1F     ;FUNC 1-4
2465  0D5B  8E 8F 90 91           BYTE  $8E,$8F,$90,$91     ;SHIFT FUNC 1-4
```

```
2469
2470                    ;
2471                    ;
2472                    ; KEYBOARD CHARACTER INTERRUPT SERVICE
2473                    ;
2474                    ;
2475 0D5F AD 09 D2     PIRQ5   LDA     KBCODE
2476 0D62 CD 00 00  E          CMP     CH1             ;TEST AGAINST LAST KEY PRESSED
2477 0D65 D0 05                BNE     PIRQ10          ;IF NOT, GO PROCESS KEY
2478 0D67 AD 00 00  E          LDA     KEYDEL          ;IF KEY DELAY BYTE > 0
2479 0D6A D0 38                BNE     PIRQ45          ;IGNORE KEY AS BOUNCE
2480 0D6C AD 09 D2     PIRQ10  LDA     KBCODE          ;RESTORE AC
2481 0D6F C9 9F                CMP     #CNTL1          ;TEST CONTROL 1 (SSFLAG)
2482 0D71 D0 0F                BNE     PIRQ20
2483 0D73 AD 00 00  E          LDA     KEYDIS          ;KEYBOARD DISABLED?
2484 0D76 D0 45                BNE     PIRQ40          ;YES.
2485 0D78 AD 00 00  E          LDA     SSFLAG
2486 0D7B 49 FF                EOR     #$FF
2487 0D7D 8D 00 00  E          STA     SSFLAG
2488 0D80 B0 3C                BCS     PIRQ40          ;(UNCONDITIONAL) MAKE -1 INVISIBLE
2489 0D82 29 3F        PIRQ20  AND     #$3F            ;MAKE OFF SHIFT AND CTRL BITS.
2490 0D84 C9 11                CMP     #HELP           ;HELP KEY?
2491 0D86 D0 0D                BNE     PIRQ25          ;NOT HELP KEY.
2492 0D88 AD 00 00  E          LDA     KEYDIS          ;KEYBOARD DISABLED?
2493 0D8B D0 31                BNE     PIRQ40          ;YES.
2494 0D8D AD 09 D2             LDA     KBCODE
2495 0D90 8D 00 00  E          STA     HELPFG          ;HELP KEY -- STORE CODE IN FLAG
2496 0D93 D0 29                BNE     PIRQ40          ;(JMP)
2497 0D95 AD 09 D2     PIRQ25  LDA     KBCODE
2498 0D98 C9 84                CMP     #CNTLF2         ;CONTROL FUNC 2?
2499 0D9A D0 1C                BNE     PIRQ30          ;NO
2500                    ;
2501 0D9C AD 00 00  E          LDA     KEYDIS          ;KEYBOARD DISABLED?
2502 0D9F D0 3F                BNE     PIRQ50          ;YES.
2503 0DA1 AD 01 D3             LDA     PORTB           ;SWITCH MODULATOR SWITCH
2504 0DA4 29 FD                AND     #$FD
2505 0DA6 8D 01 D3             STA     PORTB
2506                    ;
2507 0DA9 AD 00 00  E          LDA     SDMCTL          ;SAVE DMA STATE &
2508 0DAC F0 32                BEQ     PIRQ50
2509 0DAE 8D 00 00  E          STA     DMASAV
2510 0DB1 A9 00                LDA     #0              ;  ..DISABLE THE DMA.
2511 0DB3 8D 00 00  E          STA     SDMCTL
2512 0DB6 F0 28                BEQ     PIRQ50          ;(JMP).
2513 0DB8 8D 00 00  E  PIRQ30: STA     CH
2514 0DBB 8D 00 00  E          STA     CH1
2515 0DBE A9 03        PIRQ40: LDA     #3
2516 0DC0 8D 00 00  E          STA     KEYDEL          ;INITIALIZE KEY DELAY FOR DEBOUNCE
2517 0DC3 A9 00                LDA     #0              ;CLEAR COLOR SHIFT BYTE
2518 0DC5 85 0D        E       STA     ATRACT
2519 0DC7 AD 00 00  E  PIRQ45: LDA     KRPDEL
2520 0DCA 8D 00 00  E          STA     SRTIMR
2521 0DCD AD 00 00  E          LDA     SDMCTL          ;DMA INHIBITED?
2522 0DD0 D0 0E                BNE     PIRQ50          ;NO.
```

```
 2523                                ;
 2524  0DD2  AD 00 00      E        LDA     DMASAV      ;YES -- RESTORE DMA
 2525  0DD5  8D 00 00      E        STA     SDMCTL
 2526  0DD8  AD 01 D3               LDA     PORTB       ;ENABLE ATARI IN SWITCHBOX
 2527  0DDB  09 02                  ORA     #$02
 2528  0DDD  8D 01 D3               STA     PORTB
 2529                                ;
 2530  0DE0  68            PIRO50: PLA
 2531  0DE1  40                    RTI
 2532                                ;
 2533  0DE2                        END

ASSEMBLY ERRORS =     0
```

CROSS REFERENCE

LABEL		VALUE	REFERENCES							
ACH		0081	0							
ADRSSR	E	0008	498	531	587	589	591	603	604	607
			609	710	713	717	719	757	813	817
			884	909	911	1153	1155	1156	1159	1160
			1356	1358	1366	1383	1386	1388	1370	1374
			1387	1390	1392	1394	1396	1399	1400	1402
			1414	1417	1419	1421	1739	1778	1779	1795
			1797	1799	1801	1805	1810	1815	1821	1824
			1943	2034	2038					
ALLPOT		02DA	0							
ALLCAT	P	08AD	558	-2220						
AMFLAG		07FF	0							
AMIWIT		07F6	0							
AMIGRA		07FA	0							
AMNAME		07F9	0							
AMVTAD		07E8	0							
ANCUNV	P	0805	384	-2082						
ANTIC		D400	0							
ATTENS		0201	0							
ATTNMT	E	00AE	458	1369	1373					
ATAINF	E	0013	442	769	776	793	832	852	875	879
			886	948	1082	1093	1095	1197	1669	1955
			2185	2196	2200					
ATAINT	P	0C9F	803	-2295						
ATASCI	P	0C97	471	474	-2405					
ATRACT	E	005A	462	2518						
AUDC1		0201	0							
AUDC2		0203	0							
AUDC3		0205	0							
AUDCA		0207	0							
AUDCTL		0208	0							
AUDF1		0200	0							
AUDF2		0202	0							
AUDF3		0204	0							
AUDF4		0206	0							
B192HI		0080	0							
B192LO		0020	0							
B400HI		0085	0							
B400LO		00CC	0							
BADTCC		00BA	0							
BADMOD		0091	306							
BEEPFN		000A	0							
BEEPPP		0008	0							
BEEPNN		001E	0							
BEEPNP		0019	0							
BELL	P	060A	660	-1323	2372					
BELLI	P	060C	-1324	1326						
BICOUN	P	017E	679	-682						
BITCLR	P	07F4	1223	1621	-1629					

```
BITTON  P 0ACB      -1500  1620  1629  1639
BITUBI  P 0B12       1481 -1448
BCYUBI  P 0B02       1311  1471 -1629
BITMSK  E 0033        459  1406  1625  1630  1642
BITMSO  P 0765       1309 -1621  1718  1727
BITRGL  P 09DC      -1415  1745
BITSET  P 0788       1318 -1623
BOTBCK  E 0000        440   515   526  1113  1124  1267  1274  1667
                     1726  1968
BRKADT    0080        929  1564
BRKSET  E 0049        434   930  1001  1365  1966
BS      P 0504      -1180  2358
BS1     P 0529       1184 -1199
BS2     P 0520       1192  1194 -1196
BS3     P 0513       1187 -1189
BSA     P 050A      -1183
BUFCNT  E 0043        453   844   868   870  1875  1886  1889  1897
                     1906
BUFSTR  E 0046        456   847   849   864   866  1173  1177  1804
                     1908  1871  1873  1919  1921
CART      BFFC          0
CARTAD    BFFE          0
CARTCS    BFFA          0
CARTFG    BFFD          0
CASET     0060          0
CASSET    0040          0
CH      E 0000        437   466   923   932   937  2513
CHI     E 0021        445  2476  2514
CHACT   E 0004        438   512
CHACTL    0401          0
CHAR    E 0012        441   763   804   806  1692  1866
CHBAS   E 0003        437   509  1060  1075
CHBASE    0A09          0
CHBDT   P 0616      -1333  2383
CHKERR    008F          0
CHKLOC    4000          0
CHLEFT  P 061C      -1336  2385
CHOME   P 0613      -1331  2381
CHRIGT  P 061F      -1338  2387
CLICK   P 0A21        942  1031  1324 -2008
CLICK1  P 0A24      -2010  2016
CLICK2  P 0A2A      -2012  2013
CLOSE     000C          0
CLRLII  P 0898      -1739  1741
CLRLIN  P 08BC       1288 -1731
CLRSC2  P 04DD      -1156  1158  1162
CLRSC3  P 04EC      -1164  1167
CLRSCR  P 04D4        740   773 -1152  2356
CLRTAB  P 054E      -1222  2368
CLRTBS  P 0070       -537   539
CLS       0070          771
CNTLI     009F         2481
CNTLF2    0084         2498
```

```
CNTRLS  P 0C53          904   908   910  1954 -2045
COLAC   E 0058          461  2098  2133  2135  2196  2158  2162
COLBK     D01A            0
COLCR   F 0A35         1171  1480 -2023
COLCR1  F 0A3F         2025 -2026
COLCR2  F 0A41         2027 -2029
COLCRS  E 0047          456   848   867  1131  1132  1137  1142  1143
                       1176  1189  1190  1204  1285  1313  1403  1405
                       1425  1442  1444  1465  1456  1492  1651  1556
                       1559  1732  1859  1874  1883  1902  2029  2058
                       2060  2167  2168  2171  2173  2175  2177
COLINC  E 002D          448  2064  2086  2165
COLORO  E 0009          439   543
COLOR4  E 000C          440   586
COLPF0    D016            0
COLPF1    D017            0
COLPF2    D018            0
COLPF3    D019            0
COLPM0    D012            0
COLPM1    D013            0
COLPM2    D014            0
COLPMG    D015            0
COLRTB  P 0C4E          542 -2336
COLUMN  P 0C0B         1447  1458 -2291
COMPLT    0043            0
COMREI  F 04C2         1135 -1138  1145  1146
COMRET  P 04A4         1114 -1118  1125
CONSOL    D01F         2010
CONVR0  P 066D         1395 -1397
CONVR1  P 0672        -1399  1402
CONVR2  P 0583        -1408  1411
CONVR3  P 0689         1407 -1412
CONVR4  P 068F         1413 -1415
CONVRT  P 0656          756   805  1255  1266  1314 -1384  1733  1938
COUNTR  E 0059          462  1793  1796  1804  1807  1809  1828  1830
                       1834  2111  2121  2129  2130  2203  2205  2206
                       2208
CR        009B          777   853   878  1002
CRETRI    000D            0
CRLOOP  P 0255         -789   792
CRSINH  E 0007          438   533   825
CRSRDN  P 04A7        -1122  2350
CRSRLI  P 04BE         1133 -1136  1338
CRSRLF  P 04B4        -1131  1189  1894  1901  2352
CRSROR    008D         1570
CRSRRI  P 04CF        -1147  1336
CRSRRT  P 04C5        -1142  1203  2354
CRSRUP  P 049A        -1113  1195  1334  2348
CTIA      D000            0
CTIMNI    0000            0
CTIMLO    0002            0
DBDDEC  P 0622          606 -1348
DDDEC   P 062E          398 -1359
```

```
DBSUB   P 0A30        550   582  1049 -1280
DBSUB1  P 0A4C       1367 -1369
DBUFR1    0014          0
DELCH1  P 058C      -1255  1272
DELCH2  P 0583       1265  1268 -1273
DELCHR  P 0587      -1254  2374
DELETE    0021          0
DELLI1  P 0500      -1297  1308
DELLI2  P 05F8      -1312
DELLIA  P 0507      -1294
DELLIB  P 050B      -1296  1318  1947
DELLIN  P 0504      -1293  2064
DELTAC  E 0057        461  2080  2083  2087  2089  2090  2092  2093
                     2095  2105  2110  2114  2154  2157
DELTAR  E 0055        460  2068  2072  2076  2115  2117  2135
DELTI1  P 09D0      -1943  1946
DELTI2  P 09D7      -1947
DELTID  P 0985      -1922  1937  1944
DELTIA  P 0986       1276 -1930
DELTIB  P 098E       1932 -1934
DELTIM  P 09C1       1188 -1935
DERR    P 01E1        507  -731
DERROR    0090          0
DHLINE  P 0BF7       1398 -2282
DINDEX  E 0035        451   503   553   617   659   675   823  1397
                     1446  1458  1484  1537  1544  1653  2026
DIRECT    0002          0
DISK      0044          0
DISKID    0031          0
DISPLY    0053          0
DIV2TB  P 0C2A       1406  1423 -2312
DLISTH    D403          0
DLISTL    D402          0
DMACTL    D400          0
DMASAV  E 0027        447  2509  2524
DMASK   E 0011        441   758   811   813  1429
DMASKT  P 0C3B       1428 -2321
DNACK     008B          0
DOBI    P 0969       1879 -1881
DOBU1A  P 0978       1882  1885 -1888
DOBUF1  P 0961      -1877  1892
DOBUF2  P 098B       1887 -1895  1907
DOBUF3  P 09A0       1904 -1906
DOBUF4  P 09A4       1896  1900 -1908
DOBUFC  P 094F        863 -1868
DOCR    P 070F       1207  1459  1464  1467 -1480
DOCR1   P 06DF       1455 -1458
DOCR1A  P 0708       1472  1474 -1477
DOCR1B  P 06F3       1462 -1467
DOCR2   P 0718      -1484
DOCR2A  P 0725       1487 -1491
DOCR2B  P 0728       1490 -1492
DOCR4B  P 073B       1502 -1504  1512
```

```
DOCR4D   F 0747          1507 -1509
DOCRWS   P 0708           779   877  1465 -1478  2362
DOLCO1   P 093A         -1946  1896
DOLCO2   P 0947          1850 -1857
DOLCOL   P 092C          1138  1199  1213  1250  1278  1289  1293  1319
                         1477  1515 -1844  1917  1934
DONCHR     00E0           309  1070
DOF030   P 008F           571  -581
DOPEN    F 0020           432  -472
DOPEN1   P 009A          -559   862
DOPEN2   P 0128           629  -637   640
DOPEN3   P 0168           655   660   662  -670   673
DOPEN4   P 0194          -693   696
DOPEN5   P 019C           677   681  -698
DOPEN6   P 003A           505  -508
DOPEN7   P 01F9           739  -744
DOPEN8   P 0078          -542   545
DOPEN9   P 01E8           730  -737
DOSS     P 0320           855  -891
DOUBLE     0044             0
DRAW     P 0A4D           432 -2046
DRAW1    P 0ABF          2069 -2077
DRAW10   P 0B9F          2132  2209 -2211
DRAW11   P 0AF4          2131 -2133
DRAW2    F 0A87          2084  2094 -2096
DRAW3    P 0ADE          2113  2116 -2120
DRAW3A   P 0ABD         -2099  2104
DRAW4A   P 0AEB         -2129  2210
DRAW5    P 0AFF          2137 -2139
DRAW5A   P 0B0D          2142 -2146
DRAW6    P 0B1A          2141  2145 -2152
DRAW6A   P 0B33          2161 -2165
DRAW6B   P 0B48          2166 -2175
DRAW7    P 0B4E          2170  2172  2174  2176 -2178
DRAW8    P 0B53          2160  2164 -2180
DRAW8A   P 0B67         -2187  2198
DRAW8E   P 0B84          2194 -2199
DRAWBC   P 0B7G         -2192
DRAWR    P 0B8D          2183 -2202
DRAWW    P 0A5D          2049 -2055
DRAWB    P 0A5C          2051 -2054
DRAWLN     0011             0
DRETRI     0001             0
DSCTST     0080             0
DSPFLG   E 0016           442   897
DSTAT    E 0026           452   516   729   829   831   851   872   881
                         1000  1353  1361  1377  1963  1971
DTI        00FA             0
DTA        00FC             0
EGETC1   P 028E          -850   861
EGETC2   P 02DC           854  -862
EGETC3   P 02EA           845  -868
EGETC5   P 02FF           869   871  -877
```

```
EGETC5  P 0209        859   -861
EGETC7  P 02EE       -870    873
EGETCH  P 02AC        455    840
ENDPT   E 0032        459   1522   1523   2112   2122   2124   2140   2144
                     2159   2160
EOFERR    0088        899
EOL       009B          0
EOPEN   P 0027        437   -449    732   1539
EOT       00FE          0
EOUTC5  P 0331        893   -897
EOUTCE  P 0328       -894    899
EOUTC7  P 0347        903   -908
EOUTCH  P 0312        433   -886
ERANGE  P 0766        843    888  -1334
ERETN   P 032E       -896
ERROR     0045          0
ESCAPE  P 0494      -1107   2346
ESCFLG  E 0015        442    894    898    900   1108
EXTEN1  P 086F      -1711   1722
EXTEN4  P 0883       1720  -1723
EXTEND  P 086C       1283  -1709
FALSE     0000        494
FILDAT  E 0024        446   2195
FILFLG  E 001F        445   2055   2182
FILLIN    0012          0
FINE    E 0029        447    524    628    661   1748
FKDEF   E 005C        462    480    486   1046
FNCNOT    0092          0
FOMAT     0021          0
FORMAT    0022          0
FRMADR  E 004C        457   1677   1680   1691   1697   1700   1701   1704
FRMERR    008C          0
FUNCKY  P 0D57        479    482  -2467
GETCH   P 0204        432   -750    874
GETCHR    0007          0
GETDAT    0040          0
GETPLT  P 0213        751   -756    821   1228   1241   1269   1895   2193
GETREC    0005          0
GPRIOR  E 000D        439    564    587
GRACTL    D01D          0
GRAFM     D011          0
GRAFPD    D00D          0
GRAFP1    D00E          0
GRAFP2    D00F          0
GRAFP3    D010          0
HDR       00FB          0
HELP      0011       2490
HELPFG  E 0026        447   2495
HITCLR    D01E          0
HITONE    0005          0
HMASK   P 0C4A       1424  -2331
HOLD1   E 003A        453    555    670    689    693    702   1294   1847
                     1848   1865   1918
```

```
HOLD7   E 001B          484
HOLD3   E 0010          484  1498  1509  1613
HOLD4   E 0020          443  2186  2199
HOLDCH  E 004A          457   935  1040  1079  1089
HOME    P 04F4        -1171  1231  1333  1569
HPOSM0    000A            0
HPOSM1    0009            0
HPOSM2    0006            0
HPOSM3    0007            0
HPOSP0    0000            0
HPOSP1    0001            0
HPOSP2    0002            0
HPOSP3    0003            0
HSCROL    0404            0
ICAX1Z  E 0034          451   499   501   622   737   926
ICAX2Z  E 0033          451   492
ICCOMZ  E 0054          459  2047
INATAD  P 0825         1657 -1660
INATAI  P 0834         1653  1659 -1669
INATAC  P 0E18          752 -1652
INC2A   P 0607         1448 -1454
INCRET  P 0503        -1432  1450  1452  1453  1457
INCRSI  P 0755         1466  1493  1496  1498 -1515
INCRS2  P 06C4         1443 -1446
INCRSA  P 06DC         1225 -1443  2189
INCRSB  P 0A04          753  1262 -1437  1888
INCRSC  P 068A         1438 -1440
INCRSP  P 0A88          782 -1439
INIMLH    0007            0
INIMLL    0000            0
INSCH1  P 056E        -1239
INSCH3  P 057A         1238 -1245
INSCH4  P 0560        -1232  1244
INSCH5  P 0586         1247 -1250
INSCH6  P 0570        -1246  1249
INSCM6  P 0552        -1227  2376
INSCLR    0020            0
INSDAT  E 0040          497  1229  1239  1242  1440  1463  1473  1479
                       1497
INSLIA  P 05C1        -1283  1476
INSLIN  P 0500        -1282  2366
INTATA  P 0C93         1668 -2400
INTCHP    00CC         1062  1065
INVFLG  E 0017          443   967   969  1094
IOCBSZ    0010            0
IOCFRE    00FF            0
IRQEN     020E          521
IRQST     020E            0
JSRIND  P 030F         -884   913
KBCODE    0209         2473  2480  2494  2497
KRPDE     0048            0
KEYDEF  E 0058          462  -172   478   947
KEYDEL  E 0022          446  2478  2516
```

```
KEYDIS  E 0036          -447   840  -949  1012  1014  2482  2492  2501
KGETCH  P 033F           434   850  -924   934
KGT000  P 035A         -923   949  -955   962   970   980   987   994
                       1033  1076
KGT002  P 038A          929   941  -943  1099
KGT007  P 0391         -948  1047
KGT008  P 039D          950  -957
KGT010  P 03A3          958  -961
KGT020  P 03B5          956  -972
KGT030  P 03C3          973  -982
KGT040  P 03C9          976  -985
KGT050  P 03D0          983  -989
KGT060  P 03DC          990  -995
KGT070  P 03E2          931 -1000
KGT075  P 03E6          928 -1002
KGT080  P 03E8          997 -1005
KGT090  P 0401         1006 -1018
KGT110  P 0408         1019 -1023
KGT112  P 0414         1015 -1029
KGT115  P 0419         1029 -1038  1038  1052
KGT120  P 041C         1024 -1035
KGT125  P 042C         1041 -1045
KGT130  P 0432         1036 -1049
KGT170  P 0441         1050 -1057
KGT175  P 0458         1063 -1070
KGT177  P 045F         1068 -1074
KGT180  P 0468         1058 -1078
KGT800  P 0468          959 -1079
KGT810  P 0483         1081  1084  1086  1088 -1091
KGT820  P 048E         1003  1055 -1095
KGT830  P 0491         1092 -1096
KRPDEL  E 002E          449  2519
LEDGE     0002            0
LFRTCM  P 04C0        -1137  1148
LMARGN  E 003E          454   742  1134  1147  1183  1186  1205  1284
                       1312  1731  1736  1891  1899  1920  1931  1936
                       1941  2028
LNBUG     0001            0
LNIRQ     6033            0
LNNMI     8351            0
LO1GET  P 0804        -1637  1712  1849
LO2GET  P 0805         1299 -1638
LOCAFL    0023            0
LOGCOL  E 0048          456   857  1172  1182  1210  1217  1222  1233
                       1237  1260  1264  1441  1460  1845  1851  1854
                       1858  1860  1869  1890  1898  1909  1930  1935
LOGGET  P 0802         1208  1316 -1636
LOGMAP  E 0018          443  1164  1309  1311  1510  1615  1616  1617
LOTONE    0007            0
MOPF      0000            0
MOPL      0008            0
MIPF      0001            0
MIPL      0009            0
```

```
M2FF            0002            0
M2FL            000A            0
M3FF            0003            0
M3FL            0008            0
MASKTB        P 0C42          1605  -2326
MAXDEV          0021            0
MAXIOC          0080            0
MEMORY        M 0000            0
MEMTOP        E 0001          437   718   720
MIAOK         P 010E          619  -622
MLTTMP        E 004F          498  1385  1409  1426
MOTRQO          0034            0
MOTRST          000C            0
MOV020        P 0849         -1686  1705
MOV000        P 0851         -1691  1694
MOV090        P 0837         -1670  1688
MOVLIN        P 0838          1287 -1675
MXDMDE        P 0014           681 -2234
MXDMOD          0010            0
NACK            004E            0
NARG            0000            0
NBUFSZ          0028            0
NCOMHI          003C            0
NCOMLO          003A            0
NEWCOL        E 002B          448  2059  2061  2078  2081
NEWROW        E 002A          448  2057  2066
NMIEN           D40E          527
NMIRES          D40F            0
NMIST           D40F            0
NOCLIK        E 0023          446   938  1026  1028
NODAT           0000            0
NOFINE        P 0050          525  -527
NOFUNC        F 02AB          401  -833
NOMOD         P 00ED          601  -606
NONDEV          0092            0
NORMAL          004E            0
NOROWS        P 0C19         1491  1845 -2301
NOTE          P 00CB          869   573   864  -587
NOTE            0026            0
NOTRXD        P 014F          681   624  -657
NOTORN          0085            0
NUMBLE        P 0893          651   654   656 -2229
NVALID          0084         2052
OFFP90        P 070C         1585 -1588
OFFERS        P 0702          863   891 -1583
OLDADR        E 0050          498  1418  1422  1984  1987
OLDCHR        E 0044          455   822  1586
OLDCOL        E 0056          851  2079  2082
OLDROW        E 0042          455   790  2067  2101
OPEN            0003            0
OPNCOM        P 002F          497  -500
OPNIN           0004            0
OPNING          0000            0
```

```
OMNOT          0008           0
OPNTMP     E  0029          453   546   579   619   652   657
OUTCH      P  0828          433  -769
OUTCH2     P  0840         -804  -[illegible]  1202
OUTCHB     P  0845          778  -781
OUTCHE     P  0858          772  -738   895
OUTPLT     P  08AE          781  -766   787  1198  2181  2197
OVRRUN        008E            0
P0PF          D004            0
P0PL          D00C            0
P1PF          D005            0
P1PL          D00D            0
P2PF          D006            0
P2PL          D00E            0
P3PF          D007            0
P3PL          D00F            0
PACTL         D302            0
PAGETB     P  0BE6          600 -2272
PAL           D014            0
PBCTL         D303            0
PDEVN         0040            0
PENH          D40C            0
PENV          D40D            0
PHACRI     P  09EC        -1968  1971
PHACRS     P  09EA         1227  1254  1868 -1967  2184
PIA           D300            0
PIRQ10     P  006C         2477 -2480
PIRQ20     P  0082         2482 -2489
PIRQ25     P  0095         2491 -2497
PIRQ30     P  00B8         2499 -2513
PIRQ40     P  00BE         2484  2488  2493  2496 -2515
PIRQ45     P  00C7         2479 -2519
PIRQ50     P  00E0         2502  2508  2512  2522 -2530
PIRQS      P  005F          435 -2475
PLACRI     P  09F7        -1978  1981
PLACRS     P  09F5         1245  1277  1910 -1977  2201
PLOT          0050            0
PMBASE        D407            0
POINT         0025            0
POKEY         D200            0
POKMSK     E  0037          452   519   520
PORTA         D300            0
PORTB         D301         1008  1010  1066  1071  1074  2503  2505  2526
                           2528
POT0          D200            0
POT1          D201            0
POT2          D202            0
POT3          D203            0
POT4          D204            0
POT5          D205            0
POT6          D206            0
POT7          D207            0
POTGO         D208            0
```

```
PRINTR      0050          0
PRIOR       0016          0
PRVOPN      0081          0
PUPVL1      005C          0
PUPVL2      0093          0
PUPVL3      0095          0
PUTCHR      000B          0
PUTDAT      0080          0
PUTMSC    P 0A44       1152  1774 -2035
PUTREC      0009          0
PUTSEC      0050          0
RMRGNA    P 0000        431  -462
RAMSIZ    E 0081        449   467
RAMSYS      0001          0
RAMTOP    E 0032        451   468   547   556   642   583  1161  1675
                       1782
RANDOM      020A          0
RANGE     P 0774        760   775  1536  1538 -1540  2180  2192
RANGE1    P 0795       1550 -1556
RANGE2    P 0799       1555 -1558
RANGE3    P 077C       1542 -1544
RDONLY      0087          0
READ        0052          0
REDGE       0027          0
RENAME      0020          0
RETUR1    P 02A2        431   754   824   826  -829  1096  1577  2003
                       2211
RETUR2    P 028F        747   774   780   783  -821   880   913
RETUR3    P 078F       1575 -1577
RIRGHI      0000          0
RIRGLN      0078          0
RIRGLP      0069          0
RLEADN      0240          0
RLEADP      01E0          0
RMARGN    E 003F        454  1136  1144  1191  1451  1541  1543  1735
                       1884  1903  1939
RNGER1    P 0786       -1572
RNGER2    P 0784       1567 -1571
RNGERR    P 07AF       1547  1548  1554  1557  1560  1561 -1569
RNGOA     P 07A2       1552 -1562
ROWAC     E 0051        458  1521  1523  1524  1526  2100  2128  2134
                       2136  2138  2139  2143
ROWCRS    E 0040        454   789   846   865  1113  1117  1122  1123
                       1175  1193  1248  1266  1295  1296  1388  1393
                       1468  1483  1492  1514  1546  1636  1684  1721
                       1723  1846  1872  1881  1893  1905  1933  1968
                       1979  1992  1995  2056  2102  2147  2149  2187
                       2191
ROWINC    E 002C        448  2063  2071  2148
RSIRGN      000A          0
RSIRGP      000B          0
SAVADR    E 003D        453   608   610   724   727  1257  1259  1271
                       1275
```

```
SAVMSC  E 0058            453   590   592   648   700  1416  1420  2035
                         2037
SBUFSZ    001D              0
SCR005  P 08AA          -1748
SCR007  P 08A9          -1751  1752
SCR010  P 08B3          -1757  1759
SCR015  P 08BA          -1761  1763
SCR018  P 08CC           1768 -1771  1772
SCR019  P 08D1           1749 -1774
SCR020  P 08F5           1798 -1801  1820  1826
SCR030  P 0902          -1809  1812
SCR040  P 0909           1790 -1814
SCR050  P 091C           1817 -1827
SCR060  P 0927          -1834  1836
SCR300  P 0806          -1781  1787
SCR310  P 08E3           1783 -1789
SCREDT    0045              0
SCRFLG  E 0019            443  1231  1246  1505
SCRMEM    0093           1376
SCROLI  P 08D4           1315 -1778
SCROLL  P 08A1           1504 -1746
SDLSTL  E 000E            440   711   713   723   726
SDMCTL  E 0005            438   513   745   746  2507  2511  2521  2525
SERIN     D20D              0
SEROUT    D20D              0
SETTAB  P 0549          -1217  2370
SEX       0000              0
SHFAMT  E 0041            454   759   807  1430
SHFLOK  E 0002            437   470   975   979   986   993  1087
SHIFT1  P 0222            760  -763
SHIFT2  P 027E            808  -811
SHIFTD  P 021B           -759   762
SHIFTU  P 0276           -807   810
SIDWAY    0053              0
SIZEM     D00C              0
SIZEP0    D008              0
SIZEP1    D009              0
SIZEP2    D00A              0
SIZEP3    D00B              0
SKCTL     D20F              0
SKRES     D20A              0
SKSTAT    D20F              0
SPACE     0020              0
SPECIL    000E              0
SRTIMR  E 0023            446  2520
SSFLAG  E 0014            442   766  2485  2487
STACK   S 0000              0
STATC     0053              0
STATIS    000D              0
STIMER    D209              0
STORE   P 0626            613   632   634   638   645   647   649   665
                          667   671   686   688   691   694   699   701
                          704   707   709   715 -1353
```

```
STORE1   P 064C      -1956
STORE2     5000        0
STRBEG   P 094A      1118 -1917
STRERR   P 0651      1372 -1376
STRDK    P 0658      1294  1362  1371  1373 -1378
SUBEND   P 0758     -1830  2181  2179
SUBTMP   E 001A       443  1380  1365
SUCCES     0001       515   830  1362
SUPERT   E 0038       447   890   902   905  1054
SUPERS   P 0C89       906 -2380
SWAP     P 0A00       842   836   876   882   897   896   914  1576
                     -1988
SWAP1    P 0A09     -1992  1999
SWAP2    P 0A1E      1990 -2003
SWPFLG   E 0039       452   522  1486  1574  1878  2000  2002  2024
TAB      P 052E     -1202  1212  3060
TAB1     P 053F      1206 -1210
TAB2     P 0546      1209 -1213
TABMAP   E 0008       439   537  1624  1626  1632  1633  1641
TIMOUT     006A        0
TINDEX   E 0006       438   530
TMPCHR   E 0043       455   812   816
TMPLST   E 001C       444
TMPROW   E 001E       445  1969  1978
TOADR    E 0040       457  1678  1682  1692  1698  1702
TONE1      0002        0
TONE2      0001        0
TR100      0010        0
TR101      0011        0
TR102      0012        0
TR103      0013        0
TRNFCD     0089        0
TRUE       0001        0
TSTCT1   P 0906     -1954  1960
TSTCT2   P 09E9      1956 -1961
TSTCTL   P 09DA       892  1091 -1953
TXTCOL   E 0010       441   743
TXTMSC   E 000A       439   549   551
TXTROW   E 000F       441   741  1994  1997
UNLOCK     0024        0
UPDMEM   P 04A2     -1117  1127
VBWAIT   P 0008      -594   596
VCOUNT     D40B       594  1761  1771  2011  2017
VDELAY     001C        0
VSCROL     D405        0
VSFLAG   E 002F       449  1751  1755  1797
WIRQHI     0000        0
WIRQLM     00BA        0
WIRQLP     0096        0
WLEADN     0480        0
WLEADP     03C0        0
WRITE      0057        0
WRQMLP     0080        0
```

```
WSIRQN      000F            0
WBIRQP      000D            0
WBYNC       D40A            0
```